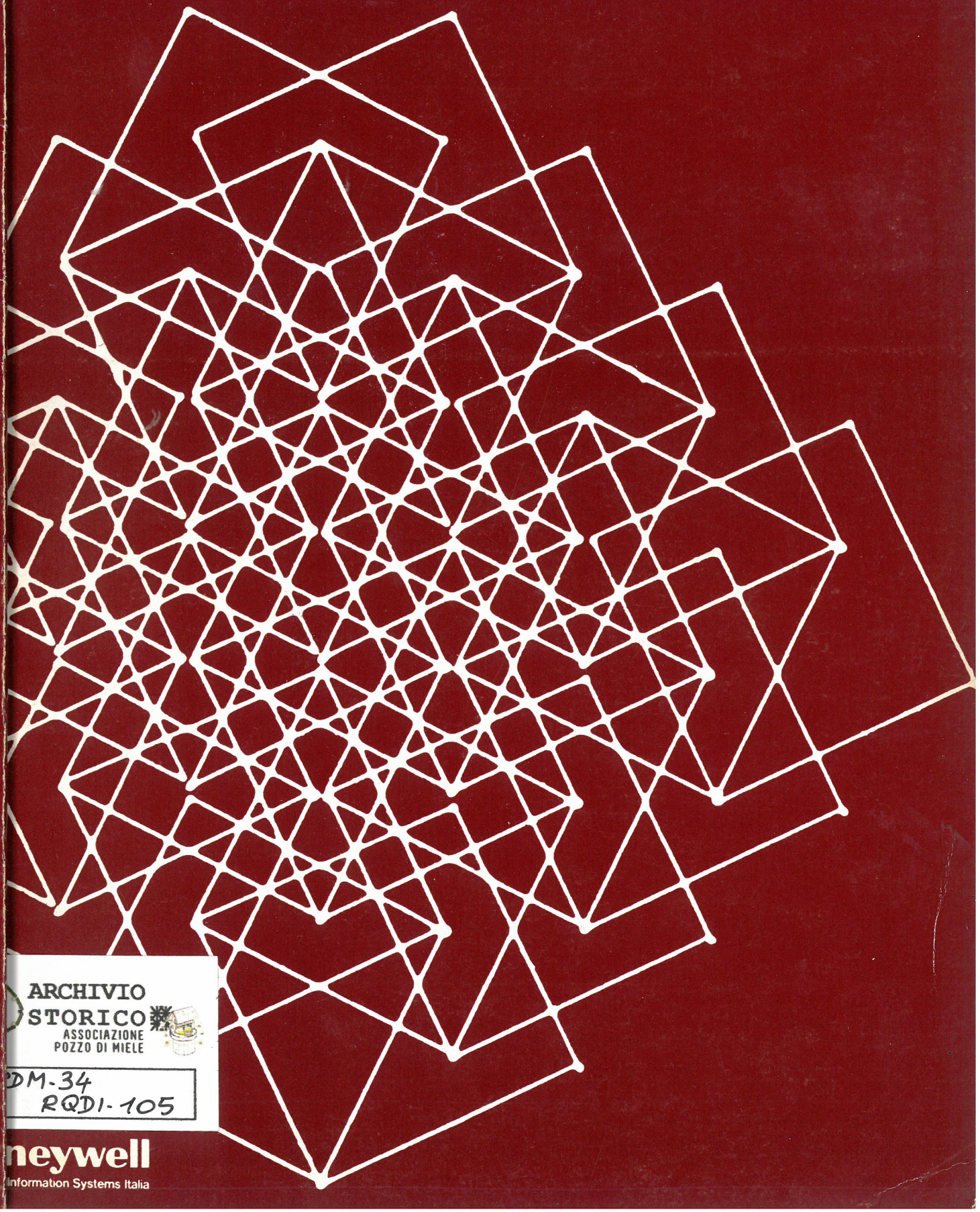
Quaderni di informatica

7

di tecnologia ed applicazioni degli elaboratori elettronici - Anno IV - Numero 1 - 1977



ARCHIVIO STORICO
ASSOCIAZIONE POZZO DI MIELE
DM-34
RQDI-105

neywell
Information Systems Italia

FPDM - 34

Quaderni di informatica

Rassegna di tecnologia ed applicazioni degli elaboratori elettronici
Anno IV - Numero 1 - 1977

Sommario

I microelaboratori
Il microelaboratore, cioè una intera, anche se piccola, unità centrale di elaborazione realizzata in una singola tessera di semiconduttore, non è soltanto un risultato spettacolare della moderna tecnologia: esso costituisce il punto di partenza per innovazioni di grande portata nella concezione dei sistemi e nelle loro applicazioni.
N. MINNAJA

Sistemi di stampa da elaboratore
In questo articolo viene completata la panoramica sui sistemi di stampa iniziata in un precedente numero della rivista, esaminando le tecniche di stampa « non a impatto ».
G. C. GATTI

Evoluzione dei sistemi di programmazione matematica e delle loro applicazioni
La « programmazione matematica » costituisce ormai un affermato strumento per la soluzione dei più svariati problemi: produttivi, finanziari, logistici. Un fattore essenziale per l'utilizzazione di questi metodi è costituito dal calcolatore elettronico.
F. RICCI

La progettazione dei sistemi informativi aziendali
La razionale utilizzazione dell'informatica nell'ambito aziendale implica una comprensione organica del « sistema azienda ». Questa ampia e multiforme problematica viene qui trattata alla luce di una moderna metodologia di analisi.
P. BLASIO

Esperienze sull'impiego degli elaboratori nell'istruzione
Vengono illustrate, attraverso la descrizione di una esperienza concreta, le possibilità di impiego del calcolatore nell'istruzione universitaria e medio superiore.
G. GAZZANIGA, L. IRONI

L'elaborazione elettronica nel settore bancario in Italia
Oltre a fornire una aggiornata panoramica statistica sull'argomento, l'articolo mette in luce interessanti correlazioni e linee di tendenza.
G. BOSISIO

Resoconti e recensioni

Direttore Responsabile
F. Filippazzi

Comitato di Redazione
A. Cicu, M. De Marco, S. Fabris, C. Falcetti, R. Levrero, P. Lupo, N. Minnaja, G. Occhini, G. Rapelli, F. Ricci, M. Rossi

Redazione
Honeywell Information Systems Italia
Centro di Ricerca e Progettazione
20010 Pregnana Milanese (Milano)

Stampa
La Tipocromo - Milano

Autorizzazione del Tribunale di Milano
n. 93 del 20 Marzo 1974

« Quaderni di Informatica » ospita articoli e contributi di varia provenienza. La responsabilità delle opinioni espresse rimane agli Autori.

La riproduzione di articoli della rivista, o di parte di essi, è consentita solo citando la fonte e l'autore.

I microelaboratori

NICOLA MINNAJA
Honeywell Information Systems Italia
Centro di Ricerca e Progettazione
Pregnana Milanese

1. Introduzione

È ormai tradizionale, facendo la storia dell'evoluzione degli elaboratori elettronici, parlare di diverse generazioni, ognuna contraddistinta da una tecnologia: per le funzioni logiche, si è passati dalle valvole ai transistori e da questi ai circuiti integrati. Per chi si fermasse a questa ripartizione, potrebbe sembrare che il progresso si sia arrestato, in quanto a tutt'oggi parliamo ancora di logica a circuiti integrati, nè più nè meno che una decina di anni fa. Se però confrontiamo con un po' più di attenzione le funzioni che vengono svolte da ogni singolo componente, ci accorgiamo subito che sta avvenendo una rivoluzione silenziosa, forse più dirompente di quella che ha portato all'utilizzazione dei primi circuiti integrati: ora vengono svolte entro una sola unità fisica funzioni complesse, che una volta erano ripartite in un gran numero di unità diverse.

Come questo è stato reso possibile? Da una parte la tecnologia dei processi sul silicio si è raffinata. Un parametro per giudicare la bontà della tecnologia è il numero di difetti per unità di superficie che si presentano alla fine della lavorazione: è intuitivo riconoscere che, se nel corso del tempo questo numero si abbassa, è possibile fabbricare con buona resa circuiti che abbiano una superficie via via più elevata, e quindi corrispondenti a funzioni logiche sempre più complesse. Dall'altra parte si sono raffinate anche le regole di tracciamento, si è trovato il modo di partire da blocchi logici elementari sempre più piccoli ed a dissipazione sempre inferiore, per cui nella stessa area di silicio si sono potute raccogliere progressivamente più funzioni elementari.

Attraverso queste due vie complementari il numero medio di blocchi elementari contenuti entro la stessa capsula è andato aumentanto decisamente.

Per dare un significato quantitativo a questa affermazione qualitativa, riportiamo in fig. 1 la capacità dei componenti di memoria viva ad accesso casuale in funzione dell'anno di apparizione sul mercato: si può osservare un incremento di circa un fattore 4 ogni 3 anni. È inutile dire che l'area della tessera di silicio su cui questa memoria è ricavata non è assolutamente aumentata in proporzione, e inoltre, fatto ancora più stupefacente, l'area occupata da un componente di memoria sulla piastra stampata è rimasta la stessa, mentre il tempo di accesso si è drasticamente ridotto, da circa a 800 a circa 300 ns.

2. Perché il microcalcolatore?

L'aumento di complessità in una memoria è facilmente quantizzabile attraverso il numero di bit; in un circuito logico si può in modo analogo parlare del numero di funzioni « porta » elementari. Tuttavia, nella progettazione di un circuito logico a complessità elevata, occorre tener presente non solo il numero di funzioni porta raggiungibile in base allo stato dell'arte, ma anche uno schema logico che consenta volumi di produzione ampi, in grado di ammortizzare le spese di progetto.

A questo scopo si può tendere attraverso varie vie. La prima in ordine di tempo è consistita nello sviluppare circuiti « su commessa », cioè che soddisfacessero esattamente le richieste di un particolare cliente, il quale praticamente finiva per addossarsi tutte le spese del progetto logico e della sua traduzione in una successione di maschere per le diffusioni richieste nel processo produttivo.

La seconda ha portato alla realizzazione di circuiti polivalenti, che possono venire specializzati dai singoli utenti specificando soltanto la maschera finale, corrispondente all'ultimo livello di metallizzazione, o eventualmente le ultime due o tre maschere, se sono ammessi più livelli di metalizzazione; una variante a questa soluzione è che la specializzazione si possa fare mediante processi diversi, da applicarsi a componente già incapsulato, e quindi possa essere demandata direttamente all'utilizzatore.

La terza via si è basata sulla ricerca di complessi di funzioni così generali, che il mercato potenziale non

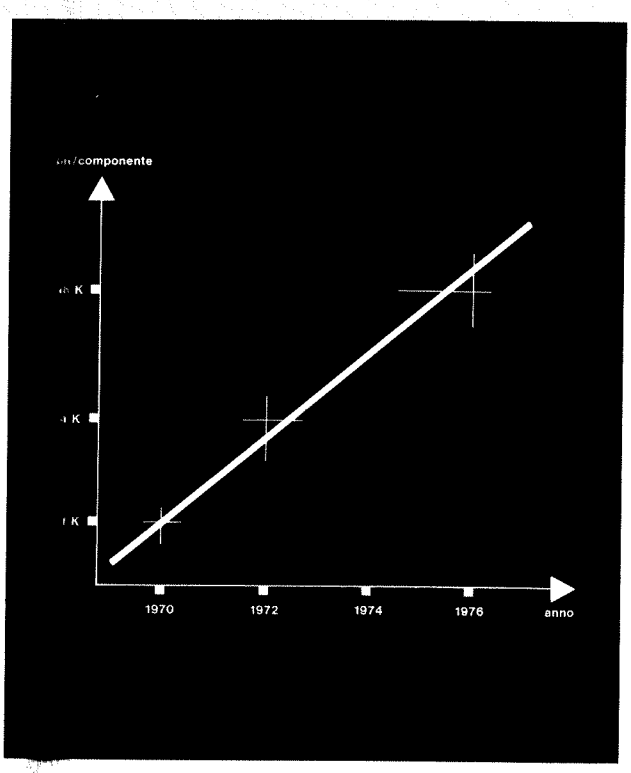


Fig. 1 - Complessità delle memorie ad accesso casuale, con tecnologia MOS dinamica

si restringesse ad uno o pochi clienti; evidentemente il successo di questo modo di procedere è strettamente legato alla bontà della scelta di queste funzioni, ed è stato abbastanza spontaneo cominciare con serializzatori e parallelizzatori da usare in trasmissione su linea sincrona od asincrona, con modem integrati, con registri tampone multipli, con calcolatori tascabili.

A questo punto, come sintesi di queste tendenze scaturite dalle esigenze dei costruttori, che automaticamente diventano esigenze degli utilizzatori in quanto si traducono in contenimento dei prezzi, nasce il microcalcolatore. Si tratta di un sistema minimo, composto di pochi componenti ad integrazione decisamente spinta, il quale è in grado di eseguire programmi. Si ripete così, su un livello di complessità più basso, la stessa rivoluzione portata nel campo della strumentazione dell'avvento dei minicalcolatori: una struttura abbastanza indifferenziata, ma estremamente flessibile, viene specializzata da un programma. È chiaro che l'esperienza già acquisita nel campo dei minicalcolatori può venire facilmente trasferita a questo nuovo campo; è anche chiaro che il livello di integrazione più alto ed il costo più basso consentono di applicare il microcalcolatore in sistemi o sottosistemi di scala troppo piccola perché la soluzione minicalcolatore fosse conveniente, e addirittura consentono di aprire per l'elettronica campi che fino a poco tempo fa le erano preclusi per ragioni di costo.

Com'è possibile questo? Una descrizione del microcalcolatore da un punto di vista strettamente sistemistico non sarebbe abbastanza esplicativa. È meglio andare a vedere quali sono i mattoni base di questo sistema minimo secondo lo schema più diffuso, quali sono le principali varianti a questo schema, quali sono le linee di sviluppo più probabili per il futuro.

3. Il cuore del sistema: il microelaboratore

Il componente fondamentale, che caratterizza il microcalcolatore, è un circuito integrato con capacità di logica e di memoria, che viene chiamato microelaboratore, o talvolta microprocessore (dal termine inglese « microprocessor »).

In realtà, l'accogliere insieme capacità di logica e di memoria nello stesso circuito integrato è una caratteristica che il microelaboratore ha in comune con altri circuiti complessi; tuttavia l'organizzazione di questa capacità ha preso un aspetto abbastanza univoco, che ora cercheremo di spiegare.

Il microelaboratore riproduce, su scala ridotta, tutte le operazioni dell'unità centrale di un calcolatore: esegue istruzioni, legge e scrive in memoria, riceve e trasmette dati attraverso circuiti di interfaccia.

La sua organizzazione è vincolata dal numero di collegamenti elettrici che esso può avere con gli altri componenti: esigenze costruttive e di spazio limitano infatti il numero dei piedini di una capsula per circuiti, mediante i quali appunto si effettuano le connessioni con l'esterno.

In ogni caso è necessario che il microelaboratore riceva un segnale di temporizzazione, il quale determina un ciclo elementare di macchina che diviene l'unità di tempo per l'evoluzione temporale dello stato interno del componente; questo segnale esterno di sincronismo è generato di solito, direttamente o tramite un circuito apposito, da un cristallo di quarzo piezoelettrico (in alcuni casi l'oscillatore è interno al componente, ma occorre qualche componente passivo per fissarne il periodo). La sospensione del segnale di temporizzazione provoca l'arresto dell'evoluzione del microelaboratore; per la maggior parte dei componenti, anzi, se questo segnale è sospeso per un tempo troppo lungo, non può essere ripresa poi l'evoluzione normale, perché non si ha soltanto una frequenza massima di funzionamento, imposta com'è naturale dai ritardi delle reti interne, ma anche una frequenza minima, imposta dalla necessaria rigenerazione periodica dello stato. Tutti gli altri segnali scambiati con l'esterno sono in relazione temporale ben precisa con il segnale (o i segnali) di sincronismo: i segnali in ingresso vengono campionati in certi momenti e per una certa durata, entro la quale devono rimanere stabili; i segnali in uscita sono validi in certi periodo, e vengono anche generati altri segnali di sincronismo per informare la circuiteria esterna che sono appunto disponibili dei dati validi. In tutto quanto diremo dopo, sottintenderemo sempre che i segnali in ingresso siano validi e stabili durante tutto il tempo richiesto.

All'inizio di ogni ciclo elementare di macchina, il microelaboratore assume un certo « stato » interno, che possiamo visualizzare attraverso il contenuto di

un certo insieme di registri, nè più nè meno che per un calcolatore di dimensioni superiori; questi registri in parte contengono dati per elaborazioni successive, in parte contengono condizioni che determinano l'evoluzione dello stato stesso. Durante ogni ciclo, in funzione dei segnali che giungono dall'esterno sui piedini di ingresso e dello stato iniziale, questo stato si modifica, e sui piedini di uscita vengono generati segnali che vanno alla circuiteria esterna.
Analizziamo più in dettaglio alcuni cicli elementari più significativi, che si presentano con caratteristiche analoghe per tutti i microelaboratori diversi.
Esiste un ciclo di macchina, il cui stato iniziale può essere sintetizzato con l'espressione « l'istruzione precedente è stata eseguita completamente »; il microelaboratore si aspetta a questo punto di ricevere un'istruzione nuova, e quindi interpreta i segnali presenti su alcuni piedini di ingresso come un codice operativo. Da quel momento, per tutti i cicli necessari, vengono eseguite in sequenza le operazioni imposte da quel codice operativo e dallo stato interno, il quale si modifica via via fino a porsi nuovamente nella condizione di « istruzione eseguita ».
L'istruzione può richiedere unicamente operazioni interne, cioè alterazioni del contenuto dei registri interni, senza scambio di dati con altri circuiti. Tipiche istruzioni di questo tipo sono i trasferimenti di dati da un registro a un altro, le operazioni logiche o aritmetiche sul contenuto di due registri ecc. In altri casi è necessario acquisire dati dall'esterno, ed allora il modo normale di procedere è in due passi: in un primo passo occorre segnalare all'esterno quale è il dato che serve, e questo avviene fornendo sui piedini di uscita un certo insieme di segnali, che possono venire interpretati come l'indirizzo della posizione di memoria che contiene il dato richiesto, oppure come il codice distintivo del circuito di interfaccia verso la periferia; in un secondo passo, dopo l'attesa necessaria perché il dato divenga disponibile, i segnali presenti su certi piedini di ingresso sono interpretati appunto come il dato richiesto, e la elaborazione prosegue. In altri casi ancora è necessario trasferire un dato presente all'interno o frutto di un'elaborazione: la sua destinazione sarà una posizione di memoria, in modo da poterlo riprendere in esame in un secondo tempo lasciando libero il registro interno in cui era conservato, oppure un circuito di interfaccia verso la periferia. Occorre allora generare sia i segnali che saranno interpretati come indirizzo sia quelli che costituiscono il dato vero e proprio da trasferire. Quello che in ogni caso è necessario, prima della fine dell'istruzione, è richiedere il codice operativo della successiva; questo codice operativo sarà registrato ancora in una posizione, di memoria, e occorrerà presentare ai piedini l'indirizzo di questa posizione, indirizzo che sarà stato elaborato durante l'esecuzione dell'istruzione precedente; a questo punto siamo di nuovo nella situazione iniziale e possiamo eseguire un'istruzione nuova.
Alcuni schemi tipici di temporizzazione sono riportati in fig. 2.

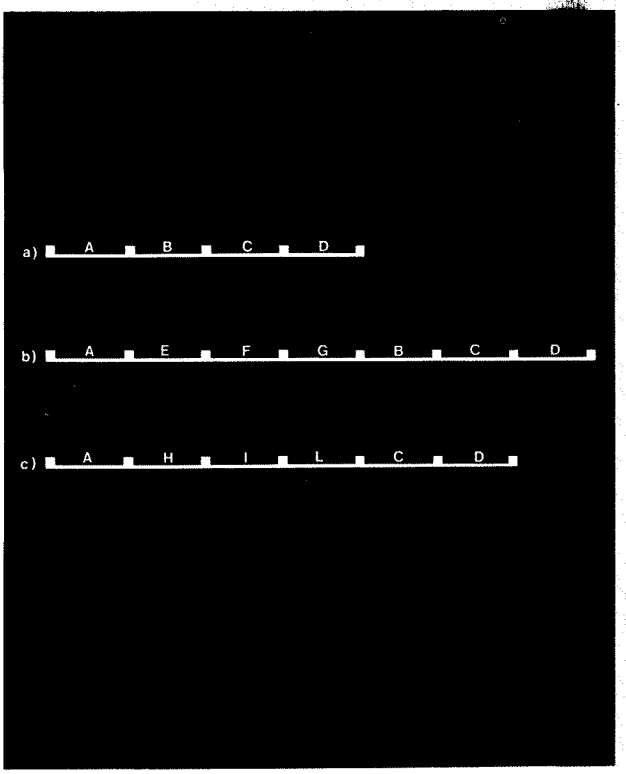


Fig. 2 - Temporizzazione tipica di un microelaboratore: a) operazioni interne; b) lettura di dati; c) scrittura di dati.
A = decodifica istruzione; B = operazione e calcolo indirizzo istruzione successiva; C = presentazione indirizzo istruzione successiva; D = lettura istruzione successiva; E = calcolo indirizzo memoria; F = presentazione indirizzo memoria; G = lettura dato; H = operazione e calcolo indirizzo memoria; I = presentazione indirizzo memoria e dato da registri; L = calcolo indirizzo istruzione successiva.

Le memoria e le interfacce

Per quanto abbiamo visto finora, il microelaboratore non va visto da solo, ma inserito in un sistema, che comprende di norma memorie e circuiti di interfaccia verso l'esterno.
Le memorie hanno ancora, per quanto abbiamo visto, una funzione duplice: contengono il programma che il microelaboratore deve eseguire e i dati su cui esso opera. Di regola il programma non viene alterato nel corso dell'esecuzione, per cui il microelaboratore non si trova mai a dover trasferire dati nella zona di memoria ad esso riservata; per garantire che nel programma non avvengano alterazioni, questo di regola viene registrato permanentemente in una memoria morta, cioè a contenuto non modificabile nel corso delle normali operazioni ma stabilito in fabbrica all'atto della costruzione. Come costruzione possiamo intendere la produzione del componente: il contenuto viene specializzato attraverso una maschera di metallizzazione; oppure possiamo intendere la fabbricazione del microelaboratore stesso: le memorie morte vengono vendute in forma indifferenziata e programmate con impulsi elettrici al momento

del loro inserimento; nei due casi parliamo rispettivamente di memorie morte in senso stretto e di memorie morte riprogrammabili, il cui contenuto può essere riportato alle condizioni iniziali di verginità da un opportuno processo, ad esempio l'illuminazione con raggi ultravioletti.
Le memorie a disposizione per i dati sono normalmente memorie a semiconduttore; in sistemi piccoli si preferiscono memorie di tipo statico, in cui cioè l'informazione permane indefinitamente; in sistemi un po' più ampi si scelgono, per ragioni di costo, memorie dinamiche, in cui l'informazione deve venire periodicamente riscritta, o « rinfrescata »: la gestione di questo rinfresco può essere affidata al microelaboratore stesso o a componenti appositi.
Lo scambio di informazioni fra il microelaboratore e le memoria avviene di regola attraverso uno scambio di segnali, che possiamo suddividere logicamente in tre gruppi; di regola questi tre gruppi di segnali viaggiano su tre gruppi indipendenti di conduttori, a cui si dà il nome inglese di « buss » (o anche « bus », traducibile in italiano come « barra » o « pista » o « corsia »).
Il primo gruppo sono i dati: la relativa corsia è impegnata dal microelaboratore, quando deve inviare dati alla memoria, o da una delle memorie, quando deve inviare dati al microelaboratore; nel caso si tratti di una memoria di controllo, questi dati saranno interpretati come un'istruzione. Il secondo gruppo costituisce l'indirizzo della posizione di memoria che deve venire registrata o letta: l'indirizzo viene generato dal microelaboratore ed inviato in parte a dei decodificatori, che selezionano il circuito di memoria interessato al trasferimento, e in parte direttamente alle memorie per la selezione della parola. Finalmente esiste una corsia di controllo, che potrebbe limitarsi, nel caso più semplice, ad un solo corrente, il quale stabilisce se si deve svolgere un'operazione di lettura o di scrittura, e che in genere serve per l'invio di tutti i segnali di controllo che servono al funzionamento del microcalcolatore.
È evidente che, nel caso semplice finora trattato, la corsia dei dati è diversa dalle altre: difatti i diversi componenti che accedono ad essa devono poter trasmettere, e tutti gli altri non devono interferire. Questo si ottiene attraverso circuiti di uscita a tre livelli, o tre stati: oltre ai due consueti livelli di segnale, deve esistere uno stato ad alta impendenza, che non altera il livello esistente sul corrente; naturalmente tutti i componenti devono trovarsi in questo « terzo stato », tolto eventualmente quello abilitato a trasmettere in quel momento. Per la corsia degli indirizzi questo non sarebbe necessario, se il microelaboratore fosse l'unico circuito che può leggere o registrare le memorie; se però si volesse ammettere anche qualche altro circuito, occorre che anche i cricuiti di uscita dell'indirizzo possano andare nel terzo stato, ed eventualmente anche i circuiti di uscita dei segnali di controllo, in modo che un altro componente possa accedere alla corsia degli indirizzi ed a quella di controllo; si dice allora che il microcalcolatore funziona con accesso diretto alla memoria, intendendo « diretto » nel senso che non c'è bisogno dell'intervento del microelaboratore.

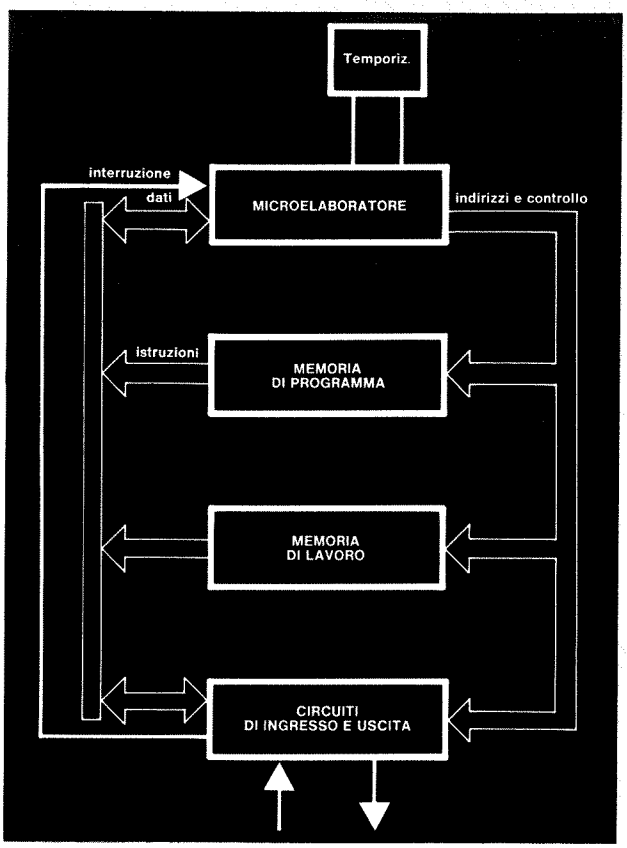


Fig. 3 - Schema di microcalcolatore

Finora abbiamo visto il sistema minimo composto dal microelaboratore e dalle sue memorie. Questo sistema deve però scambiare dati con l'esterno, e il modo più spontaneo di farlo è attraverso la corsia dei dati.
Da questa corsia saranno visibili non solo le memorie ma anche un certo numero di ingressi e di uscite, che saranno selezionabili attraverso le corsie degli indirizzi e di controllo.
Il caso più semplice sarà di ingressi che sono puramente e semplicemente porte che vengono aperte quando sono selezionate dal programma, mentre le uscite saranno circuiti staticizzatori che ad un certo comando assumono la configurazione presente sulla corsia dei dati e la conservano fino all'arrivo di un altro comando di uscita.
Il microcalcolatore assume quindi la struttura di fig. 3 con un numero di circuiti variabile a seconda della complessità del programma e del numero di segnali da scambiare con l'esterno. Questa struttura si presenta molto semplice, e non varia sostanzialmente al variare del programma: si può quindi dire che la progettazione di un sistema basato su un microcalcolatore è nettamente più semplice di quella di un sistema elettronico costruito con un minor livello di integrazione, mentre tutte le difficoltà vengono spostate nella stesura del programma.

5. I diversi tipi di microelaboratore e gli sviluppi futuri

Attualmente i microelaboratori si differenziano per il parallelismo, la velocità operativa, la capacità di memoria indirizzabile e il patrimonio di istruzioni.

Il parallelismo è il numero di bit su cui il microelaboratore opera contemporaneamente, e quindi anche il numero di correnti della corsia dei dati. Inizialmente ebbero un grande sviluppo i microelaboratori a 4 bit; ora la maggior parte è a 8 bit e ne esistono anche a 16. Il parallelismo 8 è molto diffuso perché consente di utilizzare capsule a 40 piedini senza sacrificare il numero dei segnali di indirizzo e di controllo e mantenendo separate le diverse corsie.

La velocità operativa non è definibile con pari esattezza; si può fare riferimento al tempo necessario per un'operazione interna, ad esempio la somma fra i contenuti di due registri. Questo tempo, che solo qualche anno fa era sui 20 μs, ora si aggira su 1-2 μs, in seguito ai notevoli progressi tecnologici; esistono anche componenti con architettura un po' diversa, su cui qui non ci soffermiamo, che scendono a 50-100 ns.

La capacità di memoria indirizzabile è in relazione diretta col numero dei correnti di indirizzo; per la maggior parte dei microelaboratori questo numero è 16, per cui le posizioni di memoria indirizzabili sono 65.536, capacità sufficiente per la maggior parte delle applicazioni.

Il patrimonio di istruzioni è forse la caratteristica più difficile da valutare; sono ovviamente sempre presenti le istruzioni logiche ed aritmetiche fondamentali (AND, OR, somma, complementazione), quelle di scambio di dati fra registri interni e con memoria e quelle di controllo del programma (salti condizionati); per il resto c'è un ampia varietà nella scelta delle istruzioni, legate anche al numero e alla specializzazione dei registri interni, al modo di determinare l'indirizzo di memoria a cui fare riferimento e al numero e al tipo delle condizioni che vengono controllate.

È interessante notare che la maggior parte dei microelaboratori sono sensibili ad interruzioni, cioè possono ricevere segnali che forzano un salto ad un programma particolare, eseguito il quale ritornano alla prima istruzione non eseguita del programma interrotto.

Finora i tipi diversi di microelaboratori presenti sul mercato sono parecchi, e non sembra che ci si avvii verso una unificazione; anzi, mentre all'inizio tutti i prodotti provenivano da un'unica matrice e mostravano un'evoluzione più o meno concorde verso una maggiore velocità operativa ed un maggiore parallelismo, oggi si vanno delineando quattro classi diverse di microelaboratori con caratteristiche abbastanza divergenti.

La prima classe è costituita da quei microelaboratori che sono autosufficienti o richiedono al più un altro componente per costituire un microcalcolatore completo: vale a dire che entro uno o due circuiti è contenuta tutta la capacità di memoria necessaria per le funzioni desiderate, mentre tutti i segnali da scambiare con l'esterno possono arrivare direttamente ai piedini, senza bisogno di cuircuiti dedicati. Per quanto si tratti dei componenti che consentono i costi più bassi a livello del sistema minimo, non si tratta affatto di componenti semplici, anzi di regola contengono memorie morte piuttoste ampie (almeno 8000 bit); il costo diventa allora effettivamente molto basso se la programmazione è fatta attraverso una maschera di metallizzazione, e ciò diventa economico solo se i volumi sono molto elevati. Il campo di applicazione di questi componenti è quindi essenzialmente quello dei beni di consumo, come il controllo di elettrodomestici, bilance automatiche, ascensori, calcolatrici semplici.

La seconda classe si differenzia dalla prima essenzialmente per la necessità di scambiare con l'esterno un numero di segnali molto maggiore di quanto non sia consentito dai piedini di uno o due circuiti; il microcalcolatore richiede quindi un certo numero di circuiti di ingresso e di uscita, che devono venire indirizzati separatamente. La complessità delle elaborazioni interne può essere abbastanza limitata, e le esigenze di sincronismo con gli eventi del mondo esterno possono essere contenute: tipiche applicazioni possono essere il controllo di una rete di semafori, di un concentratore di linee di trasmissione, di un'apparecchiatura di condizionamento climatico. Per questi microcalcolatori può essere usato anche un microelaboratore della prima classe, se è stato progettato in maniera da consentire un ampliamento delle interfacce verso l'esterno.

La terza classe è quella in cui si è già arrivati ad una certa maturità. La complessità e la varietà dei compiti da eseguire richiedono una capacità di memoria che sicuramente non può essere contenuta all'interno del microelaboratore; le interazioni con l'esterno devono essere ben sincronizzate, per cui è assolutamente necessaria la capacità di interruzione, preferibilmente con salto automatico ad un programma diverso a seconda dell'origine dell'interruzione stessa. Inoltre il formato dei dati che vengono scambiati con l'esterno spesso richiede una preelaborazione prima che essi siano in forma utile per la memorizzazione: ad esempio si vuole trasmettere o ricevere serialmente; per quanto il microelaboratore sia perfettamente in grado di svolgere compiti di questo tipo, essi possono richiedere un tempo di esecuzione troppo elevato, per cui sono stati sviluppati circuiti di interfaccia che siano in grado di fare eleborazioni e controlli preliminari. Un microelaboratore dialoga con questi circuiti non solo scambiando dati, ma inviando comandi e leggendo parole di stato; i comandi possibili dipendono dal particolare circuito di interfaccia e possono essere abbastanza complessi. È possibile che sia richiesta l'interazione con memorie elettromeccaniche come dischetti magnetici; come circuiti di interfaccia possono funzionare addirittura microelaboratori della prima classe.

Le applicazioni di questi microelaboratori spaziano dal governo di unità periferiche di calcolatori (stam-

panti, dischetti) a giochi televisivi, a centrali telefoniche, a controllo di mezzi di trasporto; gli sviluppi tecnologici e sistemistici hanno portato ad un progressivo aumento di prestazioni e alla definizione di parecchi circuiti periferici, non solo di interfaccia verso l'esterno, ma anche per il controllo dell'accesso diretto alla memoria, per la gestione di più livelli di interruzioni, per operazioni aritmetiche in virgola mobile, per conteggi di eventi o per misure di tempo.
La quarta classe è rivolta soprattutto al campo dei minicalcolatori, e qui è meno evidente una linea di tendenza ben precisa, anzi ne emergono due, e non è ancora chiaro se si tratterà di linee in contrapposizione, una delle quali alla lunga risulterà vincente, oppure se questa quarta classe si sdoppierà originando due classi diverse. La prima linea si basa su microcalcolatori con parallelismo 16 e con un ampio patrimonio di istruzioni; la velocità operativa è paragonabile con i microelaboratori della terza classe, ma le prestazioni risultano nettamente superiori per il parallelismo doppio e per la maggiore potenza delle istruzioni; in alcuni casi queste sono già oggi compatibili con le istruzioni di famiglie note di minicalcolatori, di cui consentono l'estensione verso il basso.
La seconda linea si basa su componenti a velocità molto elevata ma a densità d'integrazione più bassa, per cui la gestione del contatore del programma di solito è demandata a un componente diverso da quello dove hanno sede l'unità aritmetica e i registri di lavoro: inolte questi sono organizzati in gruppi limitati, ad esempio di 4 bit, con la propagazione di un riporto per consentire il parallelamento di più unità. Naturalmente questa seconda linea lascia al progettista una notevole libertà, ad esempio nel fissare il parallelismo dei dati, le condizioni di salto, le gestione delle interruzioni.

6. Conclusioni

Il microelaboratore è il risultato congiunto dei progressi nel campo dell'integrazione circuitale e di una razionalizzazione e generalizzazione nelle tecniche di progetto elettronico.
Dalla sua realizzazione sono stati resi possibili prodotti che prima avrebbero avuto un costo inaccettabile, come le calcolatrici elettroniche e i giochi su schermo televisivo; altri prodotti di larga diffusione, come le automobili, stanno per utilizzare questi componenti in maniera massiccia. Nel campo dei sistemi di acquisizione ed elaborazione di dati il microelaboratore sempre più spesso prende le funzioni di gestione di unità periferiche e, nel caso di minicalcolatore, anche di unità centrale, consentendo la progettazione di un mini secondo le specifiche esigenze di ogni utente con una specializzazione minima. L'introduzione del microcalcolatore ha anche rivoluzionato le tecniche di progettazione, in quanto una buona parte del progetto consiste nella stesura dei programmi, e ciò ha richiesto un notevole sforzo di aggiornamento tecnico. Questo fatto può avere rallentato la diffusione iniziale di questi componenti; ora tuttavia questo problema è superato e alla riduzione dei costi di produzione portata dal livello di integrazione si sta affiancando anche la riduzione dei costi e dei tempi di progetto.
Non è quindi difficile pronosticare per questi componenti un futuro brillante con campi di applicazione molto vasti.

7. Bibliografia

La letteratura sui microcalcolatori è in continuo aumento, ma per opere sistematiche bisogna ancora fare riferimento ai manuali dei diversi costruttori, alcuni dei quali trattano in modo approfondito applicazioni concrete. In italiano esiste il volume di M. Molinari « Introduzione ai microelaboratori » (Edit. Il Rostro) con descrizione dei principali prodotti e con esempi di applicazioni.
Alcuni articoli di rassegna sono comparsi su un numero speciale della rivista « Alta Frequenza » (giugno 1975), con cenni a realizzazioni progettate in Italia.
Una rassegna sui principali circuiti di interfaccia è: N. Minnaja « Circuiti integrati d'interfaccia per microelaboratori », Atti della Giornata di Studio su « I circuiti integrati complessi: tecnologie e applicazioni », Gruppo di Microelettronica AEI, Genova, 21 Novembre 1975.
Una raccolta di articoli usciti via via sulla rivista « Electronics » e che hanno trattato il problema da tutti i punti di vista è « Microprocessors », Electronics Book Series, 1976; più orientato alle applicazioni sui microelaboratori della seconda e terza classe è: J. Peatman, « Microcomputer-based design », McGraw-Hill 1977.
La rivista « Euromicro » è praticamente dedicata completamente ai problemi dei microcalcolatori; nel n. 6 del 1976 c'è una bibliografia riassuntiva molto ampia.
Per gli ultimi sviluppi e le previsioni a breve termine si possono consultare gli atti della « 1977 International Solid State Circuit Conference », Philadelphia 1977, in particolare le relazioni introduttive alle due tavole rotonde su « Low end microprocessors » e « High end microprocessors » e le comunicazioni tenute nella sessione « Microprocessor advancements ».

Sistemi di stampa da elaboratore

GIANCARLO GATTI
Honeywell Information Systems Italia
Centro di Ricerca e Progettazione
Pregnana Milanese

1. Introduzione

In un precedente articolo[1] sono state illustrate le tecniche di stampa a impatto; questo secondo articolo completa la rassegna presentando i sistemi di stampa a non impatto.

Le stampanti a non impatto offrono motivi validi di interesse per i loro pregi in fatto di velocità, affidabilità e silenziosità. I metodi di stampa differiscono infatti sostanzialmente da quelli visti nell'articolo precedente per il fatto che non sono richiesti elementi percussori per scrivere un carattere; questo è infatti generato per via elettrica, ottica o magnetica. In accordo a ciò, le parti in movimento sono ridotte al minimo, col vantaggio di semplificare la complessità meccanica dell'apparecchiatura. Per contro, i sistemi di stampa a non impatto presentano determinati svantaggi. Non si possono infatti realizzare copie simultanee; inoltre diverse soluzioni richiedono l'uso di carta speciale con riflessi negativi sui costi di esercizio.

Nel seguito dell'articolo verranno descritti i principali tipi di stampa a non impatto, presentando sinteticamente i principi di funzionamento e le caratteristiche operative fondamentali.

2. Generalità sulle stampanti a non impatto

Si possono anzitutto distinguere due categorie di dispositivi: a linea e seriali (definiti come nel caso delle stampanti a impatto).

Le stampanti a linea vengono utilizzate come stampanti di uscita dell'elaboratore. Le massime velocità di stampa sono di 20.000 LPM (Linee per Minuto), velocità sette volte superiore alla più veloce stampante ad impatto. Le colonne di stampa, o il numero dei caratteri per linea stampata, vanno da 132 a 160 posizioni. In questi tipi di macchina i moduli di carta sono continui (fan fold) e la tabulazione verticale viene controllata dall'elaboratore che ha in memoria la posizione di tutte le linee da stampare sul modulo. Queste macchine presentano meccanismi di controllo e trasporto della carta assai complessi e costosi; basti pensare che occorre garantire un buon allineamento con moduli che scorrono alla velocità di circa un metro/secondo.

Le stampanti seriali (salvo un tipo particolare ad altissima velocità; v. par. 3.5) hanno impieghi simili alle telescriventi, in consolle degli elaboratori, in applicazioni dove è importante la silenziosità e dove non sia necessario disporre subito di copie dell'originale. La velocità di stampa sono nel campo di 20 ÷ 500 car/sec., in funzione delle tecniche impiegate. Le colonne di stampa variano da 80 a 132 posizioni. Possono essere inseriti, oltre ai moduli continui, fogli singoli, (single sheet). In opzione, si può avere una tastiera alfanumerica ed il dispositivo VFU (Vertical Format Unit) per il controllo della tabulazione verticale.

In fig. 1 sono mostrati schematicamente i gruppi che compongono una stampante a non impatto. Essi sono uguali o simili, tranne ovviamente quello di stampa, ai gruppi delle macchine ad impatto. Il gruppo di stampa rimane sempre il più complesso e cambia in funzione dei principi di scrittura adottati. Lo schema di funzionamento in generale è simile a quello descritto per le stampanti ad impatto. L'informazione è inviata in una memoria di transito e gli elementi di scrittura vengono attivati con opportuno sincronismo in maniera da formare il carattere selezionato nella giusta posizione lungo la linea di stampa.

Si possono stampare caratteri ad immagine intera (solid font) o per punti (dot matrix). I set grafici di caratteri utilizzati sono diversi; pochi però sono quelli omologati da ISO e ASCII. Nella classe delle stampanti a linea la stampa del carattere è fatta per punti o per intero mentre in quella seriale è solo per punti. Nelle stampanti a punti, il set di caratteri può essere, in generale, variato in modo abbastanza semplice, agendo sulla logica che controlla la formazione dei

[1] Quaderni di Informatica, n. 5, 1976.

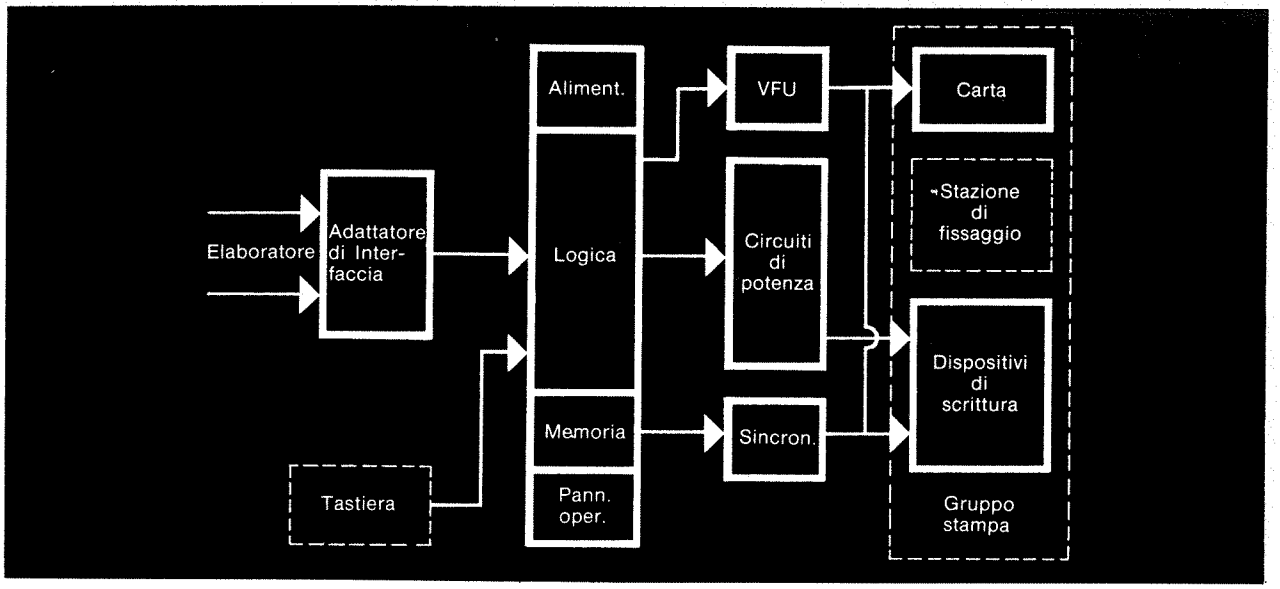


Fig. 1 - Schema a blocchi di stampante non a impatto.

caratteri; in alcuni casi, ad esempio con la stampante a laser (par. 3.5), è anche possibile variare da programma le dimensioni dei caratteri.

I principi impiegati per realizzare una stampa senza impatto sono numerosi ed i più significativi verranno esaminati nel paragrafo seguente. In termini generali si può tuttavia dire che l'immagine viene generata in uno dei seguenti modi:

— direttamente sulla carta, con visibilità immediata;
— direttamente sulla carta, ma come immagine latente; la visibilità è ottenuta mediante un successivo trattamento (sviluppo);
— indirettamente, su un supporto intermedio da cui viene poi trasferita sulla carta.

Secondo il principio di scrittura, la carta può essere di tipo normale oppure speciale. Tra queste ultime sono di particolare importanza:

— le carte con uno strato dielettrico, usate nei vari sistemi di stampa elettrostatica;
— le carte termosensibili (carte termochimiche; carte con uno strato bianco a bassa temperatura di fusione, posto su un fondo nero);
— carte porose per la stampa elettrolitica.

Il costo delle carte trattate costituisce un elemento a sfavore di determinate soluzioni, ma può essere bilanciato da altri fattori positivi da queste presentate.

3. Tecniche di stampa

In questo paragrafo vengono descritte le tecniche di stampa più significative realizzate negli ultimi anni. Le diverse soluzioni sono tra loro confrontate nella tabella di fig. 2.

3.1. *Stampa elettrografica*

È questo un sistema di stampa elettrostatica a visibilità differita, che richiede carta trattata. In figg. 3 e 4 sono schematizzati rispettivamente il tipo a linea e quello seriale.

Il carattere è costituito da una serie di punti posti in un reticolo o matrice, normalmente di 7×5 posizioni. Gli elettrodi posti davanti alla carta ad una distanza di alcuni decimi di millimetro ed oppurtunamente pilotati dalla logica elettronica della macchina, sono attivati selettivamente in corrispondenza del carattere da stampare. Si creano così sulla carta, che

Fig. 2 - Tecniche di stampa a non impatto: caratteristiche principali

Tecnica di stampa	Velocità massima	Categoria	Processo di stampa	Forma del carattere	Visibilità	Qualità di stampa	Affidabilità	Carta
Elettrografica	18.000 LPM	A linea	Con sviluppo	A punti	Differita	Buona	Ottima	Speciale
Elettrografica	500 car/sec	Seriale	Con sviluppo	A punti	Differita	Buona	Ottima	Speciale
A plasma	20.000 LPM	A linea	Diretto	A punti	Immediata	Buona	Buona	Normale
A getto di inchiostro	200 car/sec	Seriale	Diretto	A punti	Immediata	Discreta	Buona	Normale
Xerografica	8.000 LPM	A linea	Indiretto	Continuo	Differita	Ottima	Buona	Normale
Xerografia a laser	45.000 car/sec	Seriale	Indiretto	A punti	Differita	Buona	Buona	Normale
Magnetografica	12.000 LPM	A linea	Indiretto	A punti	Differita	Ottima	Ottima	Normale
Termica	100 car/sec	Seriale	Diretto	A punti	Immediata	Discreta	Discreta	Speciale
Elettrolitica	250 car/sec	Seriale	Diretto	A punti	Immediata	Discreta	Scarsa	Speciale

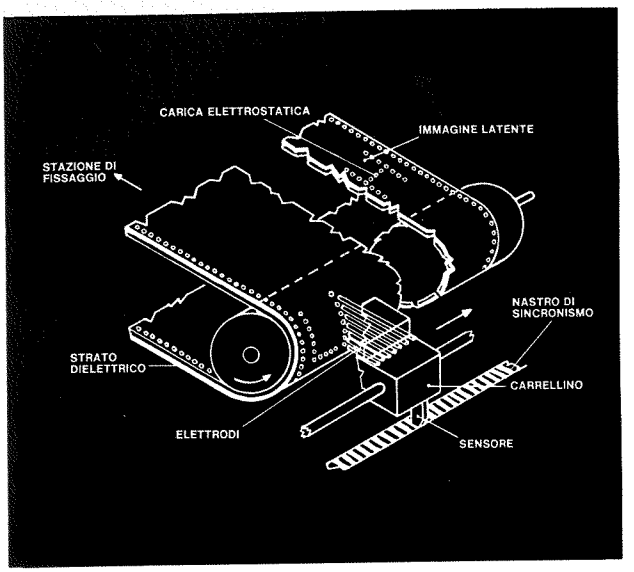


Fig. 3 - Stampante elettrografica a linea

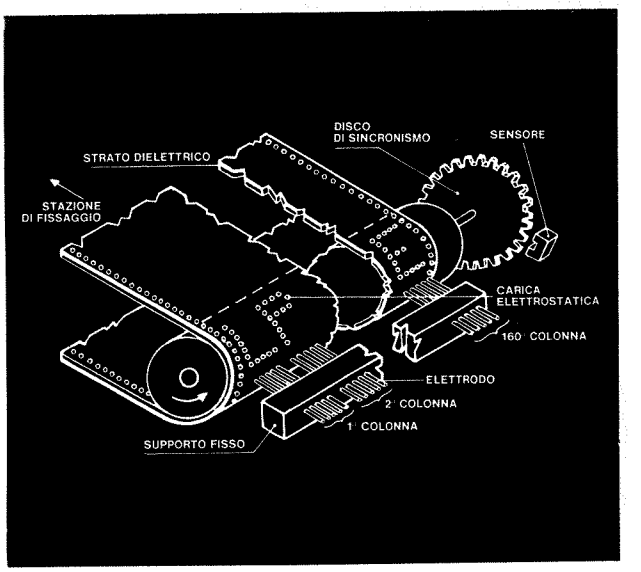


Fig. 4 - Stampante elettrografica seriale

è ricoperta con uno strato di materiale dielettrico, delle cariche elettrostatiche formanti una immagine latente nelle posizioni dove occorre stampare dei punti. Il tempo richiesto per depositare le cariche elettriche è nell'ordine dei microsecondi; le correnti in gioco sono bassissime.

L'immagine latente è resa visibile con un processo successivo, inviando sulla carta un flusso di polvere di inchiostro (toner) caricato elettrostaticamente. Le cariche presenti sullo strato dielettrico della carta, attraggono le particelle di inchiostro, che sono caricate con segno opposto. Il toner è successivamente fissato sulla carta tramite pressione e riscaldamento. Questo processo di stampa è illustrato in fig. 5.

Nella categoria a linea, gli elettrodi sono allineati e paralleli all'asse di stampa; nel caso considerato, vi sono cinque elettrodi per ogni carattere o colonna. Il moto della carta è generato da un motore passo-passo in maniera che, ad ogni impulso, la carta si sposta di 1/7 dell'altezza del carattere per realizzare i punti verticali. La velocità di queste stampanti arriva fino a 18.000 LPM.

Nella categoria seriale, la stampa è ottenuta con una fila verticale di sette elettrodi montati in una testina che trasla, su apposito carrellino, parallelamente alla linea di scrittura. La disposizione è identica a quella vista per la stampante ad impatto seriale ad aghi. La velocità massima di stampa arriva fino a 500 car/sec.

3.2. *Stampa a plasma*

Questo sistema (Plasma Jet) illustrato in fig. 6, consente la stampa diretta su carta non trattata. La macchina è a linea e la stampa del carattere avviene per punti in una matrice poniamo di 7×5. La disposizione degli elettrodi ricalca l'approccio seguito nella stampante a linea descritta nel par. 3.1. e così dicasi per il moto della carta.

Tra gli elettrodi e la carta fluisce un sottile strato gassoso costituito da particelle di inchiostro caricate negativamente. Attivando elettricamente gli elettrodi, si staccano da questi ioni positivi che attraverso lo strato gassoso proiettano le particelle di inchiostro sulla carta nelle posizioni desiderate.

Rispetto al metodo elettrografico visto nel par. 3.1., questo sistema presenta vantaggi quali l'impiego di carta normale e la semplificazione del processo di

Fig. 5 - Processo di sviluppo stampa elettrografica.

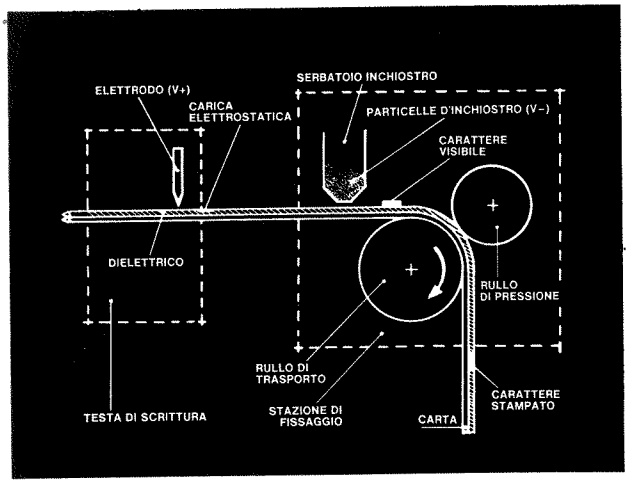
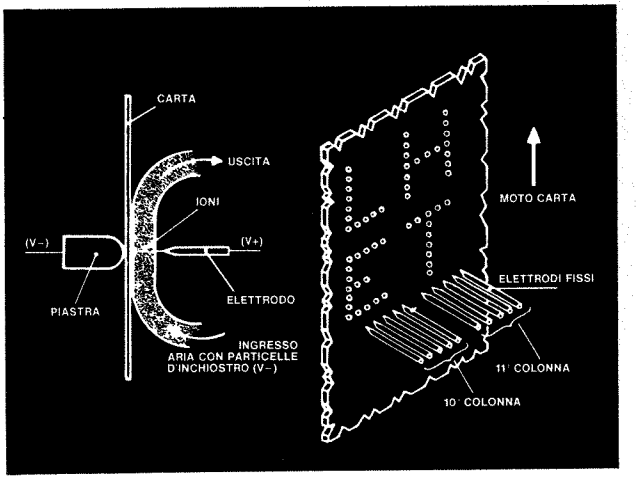


Fig. 6 - Stampante a plasma



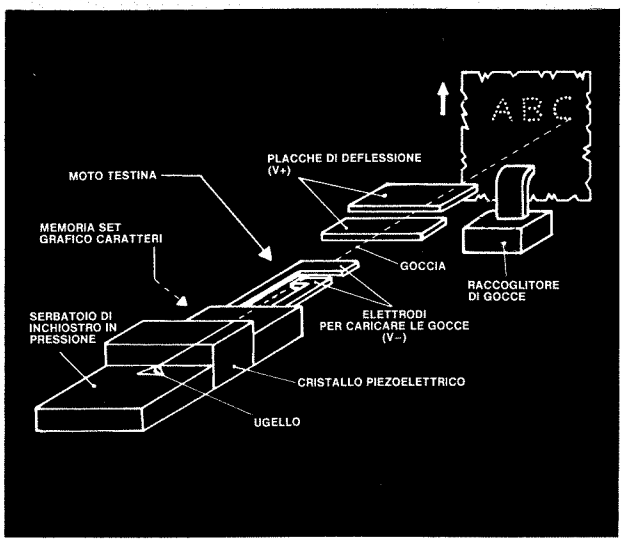


Fig. 7 - Stampante a getto di inchiostro

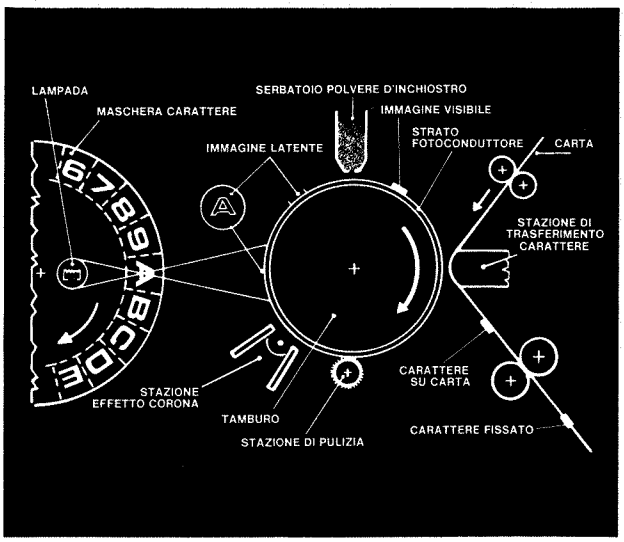


Fig. 8 - Stampante xerografica

scrittura. La velocità di stampa può raggiungere le 20.000 LPM.

3.3. *Stampa a getto di inchiostro*

In questo sistema (Ink Jet) la scrittura è realizzata mediante un sottile getto di inchiostro che descrive il carattere direttamente sulla carta. In effetti il getto è costituito da una successione di minuscole gocce di inchiostro (circa 25 centesimi di millimetro in diametro) caricate elettrostaticamente, che vengono guidate sulla carta mediante opportuni campi elettrici. Ogni goccia si comporta in modo analogo a quello di un elettrone in un tubo catodico. I caratteri sono quindi formati da una serie di punti vicinissimi fra loro così da apparire continui. La fig. 7 illustra il metodo in questione.

Un semplice circuito idraulico mantiene l'inchiostro a pressione costante entro un serbatorio di capacità adeguata. Un ugello, posto all'uscita del serbatoio, regola il flusso del getto che viene interrotto da vibrazioni ad alta frequenza generate da un cristallo piezoelettrico; in corrispondenza si formano le gocce. Queste vengono caricate elettrostaticamente in modo proporzionale alla deflessione desiderata della loro traiettoria. Le gocce attraversano, successivamente all'uscita dall'ugello, un campo elettrostatico fisso che le indirizza con angoli opportuni verso la carta. Le gocce non deviate vengono catturate da un raccoglitore e rimesse in ciclo.

Il meccanismo è posto su un carrellino che trasla davanti alla carta con moto parallelo alla linea di stampa. I punti sono posizionati mediante lo spostamento orizzontale della testina di scrittura e la deflessione verticale operata dal campo elettrico.

La velocità massima di stampa è nell'ordine di 100 car/sec.

3.4. *Stampa xerografica*

Questo sistema (Xerography) è di tipo elettrostatico a stampa indiretta su carta normale. Il funzionamento è a linea, e la stampa del carattere è del tipo ad immagine intera.

Il processo si basa sulla formazione di un'immagine latente, costituita da cariche elettrostatiche, realizzata su supporti opportuni (tamburo rotante, nastro, ecc.). Tale immagine viene formata illuminando selettivamente la superficie di materiale fotoconduttore, con cui è rivestito il supporto. Lo strato fotoconduttore deve essere preventivamente caricato elettrostaticamente mediante « effetto corona ». Nelle zone illuminate si formano coppie elettrone-lacuna che permettono la dispersione della carica superficiale.

Nelle zone non illuminate rimane la carica elettrostatica; a questa immagine latente si fa aderire una polvere di inchiostro (toner) carica con segno opposto, che viene successivamente trasferita sulla carta e fissata.

In fig. 8 è illustrato uno schema di funzionamento. Un tamburo avente sulla circonferenza delle maschere rappresentanti il set grafico dei caratteri (un set per colonna di stampa) è in rotazione continua. All'interno di esso vi sono delle lampade, che per semplicità supponiamo essere una per colonna. Allorché la maschera del carattere da stampare si presenta davanti al cono di luce della lampada, questa viene accesa per un breve tempo. L'immagine del carattere viene focalizzata sull'altro tamburo rivestito con strato fotoconduttore. Nelle aree colpite dalla luce le cariche elettrostatiche superficiali vengono dissolte mentre quelle presenti nelle zone d'ombra rimangono inalterate.

L'immagine latente viene successivamente sviluppata inviando un flusso di toner, che viene trattenuto solo dalle aree caricate elettrostaticamente. L'immagine, seguendo il moto del tamburo, passa poi attraverso la stazione di trasferimento dove il carattere viene ceduto alla carta. Successivamente, le particelle di inchiostro sono fissate sulla carta mediante pressione e riscaldamento.

Eventuali particelle rimaste sulla superficie del tamburo vengono eliminate durante il passaggio attraverso una stazione apposita. Dopo questa operazione, la superficie fotoconduttrice è pulita e pronta per ripetere il ciclo di stampa.

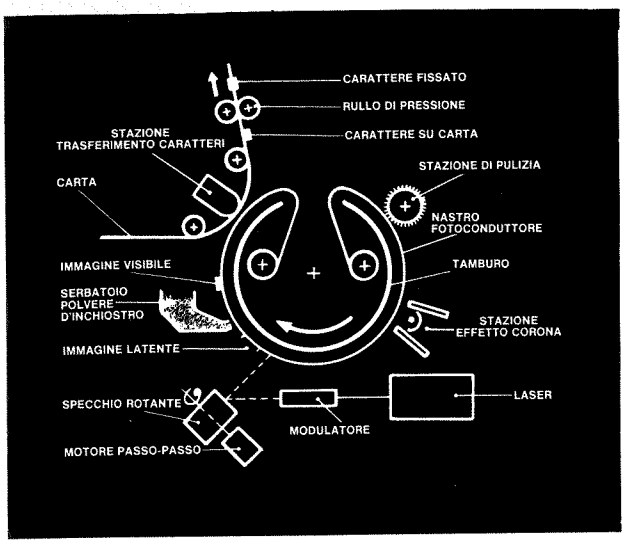


Fig. 9 - Stampante xerografica a laser

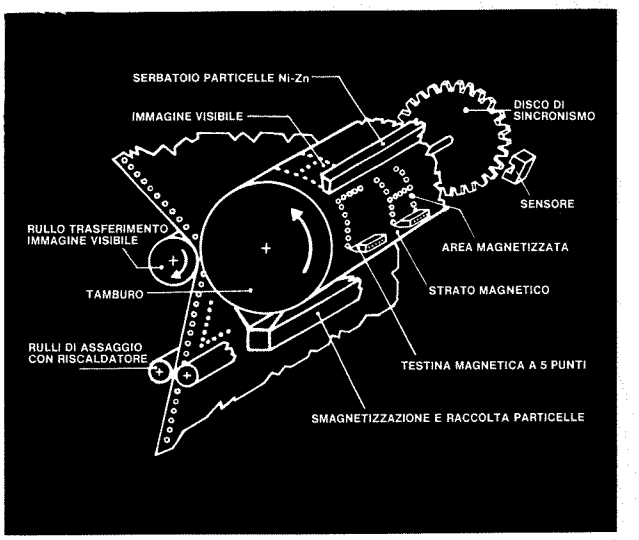


Fig. 10 - Stampante magnetografica

In queste macchine le velocità operative raggiungono le 8.000 LPM.

3.5. *Stampa xerografica a laser*

Questo sistema (fig. 9) è in sostanza una variante del sistema descritto nel paragrafo precedente.

L'organizzazione è seriale e la stampa del carattere è del tipo a punti. Si possono ottenere punti vicinissimi fra loro e la qualità di stampa risulta quindi buona e molto vicina a quella con immagine intera. Modulando il raggio laser si possono realizzare diversi set grafici di caratteri senza modificare la macchina. L'alta velocità di stampa pari a 45.000 car/sec. o 13.400 LPM, fa di questa macchina la più veloce delle stampanti seriali (impatto e non).

Un nastro fotoconduttore è teso sulla superficie di un tamburo avente una feritoia nella quale viene fatto passare per essere raccolto su bobine. Un dispositivo, non illustrato in figura, provvede a fare scorrere il nastro dopo un certo periodo di impiego, in maniera da presentare una superficie vergine per le scritture successive.

Il tamburo è in rotazione continua e la superficie del nastro fotoconduttore viene caricata elettrostaticamente nella stazione a effetto corona per essere pronta a ricevere l'immagine.

Un raggio laser a bassa potenza, opportunamente diretto in funzione del carattere da stampare, viene inviato contro uno specchio rotante e riflesso verso lo strato fotoconduttore.

Una linea di caratteri viene scritta in orizzontale mediante la deflessione procurata dallo specchio ed in verticale sfruttando la rotazione del tamburo. Data l'alta velocità di deflessione orizzontale non vi è disallineamento apprezzabile della linea di stampa.

L'immagine latente formata sul nastro fotoconduttore viene sviluppata con l'applicazione del toner e successivamente trasferita sulla carta e fissata con un processo uguale a quello descritto nel paragrafo precedente.

3.6. *Stampa magnetica*

Questo sistema, schematizzato in fig. 10, è concettualmente il duale, in termini magnetici, dei sistemi elettrografici indiretti. Il dispositivo di stampa è costituito da testine magnetiche, una per carattere, disposte lungo la linea di stampa e vicinissime ad un tamburo rotante, ricoperto di uno strato magnetico avente opportune caratteristiche. La testina è costruita in modo da magnetizzare, lungo una linea orizzontale, una serie di piccole aree corrispondenti ai punti che compongono il carattere. Struttando la rotazione del tamburo, si costruisce sulla sua superficie una immagine latente dei caratteri della linea.

L'immagine latente viene poi sviluppata applicando alla superficie del tamburo un toner speciale che aderisce solo sulle zone magnetizzate. Successivamente, l'immagine viene trasferita sulla carta e fissata, al solito, mediante pressione e riscaldamento.

Infine, una stazione provvede alla smagnetizzazione del tamburo ed alla raccolta delle particelle residue, rigenerando la superficie magnetica prima che si presenti di nuovo sotto le testine di scrittura.

La velocità di stampa ottenibile e con questo sistema sfiora le 12.000 LPM.

3.7. *Stampa termica*

Questo processo di scrittura utilizza l'energia termica per modificare lo stato di carte opportunamente trattate.

Una disposizone tipica è quella illustrata nella fig. 11. Una testina con sette microresistori è traslata lungo la linea di stampa da un motore passo-passo. I segnali elettrici di comando scaldano selettivamente i resistori; il calore da questi generato fa fondere lo strato superficiale bianco in modo da creare, per contrasto con la base nera, i punti formanti il carattere. In altre soluzioni, il calore innesca una reazione chimica che colora la carta. In ogni caso, per conseguire velocità di operazione adeguate, occorre che gli elementi di scrittura presentino una bassa inerzia termica.

Una variante di queste tecniche di stampa si può considerare quella a scarica elettrica: la tensione applicata ad elettrodi puntiformi, posti a contatto con la carta, genera un arco di corrente che incide lo

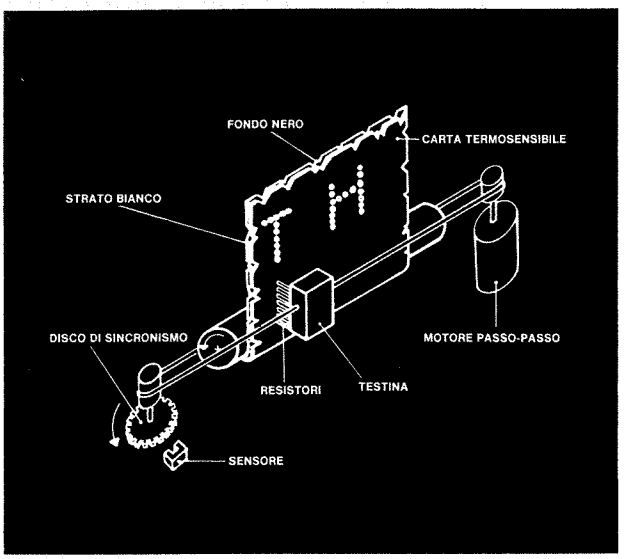


Fig. 11 - Stampante termica

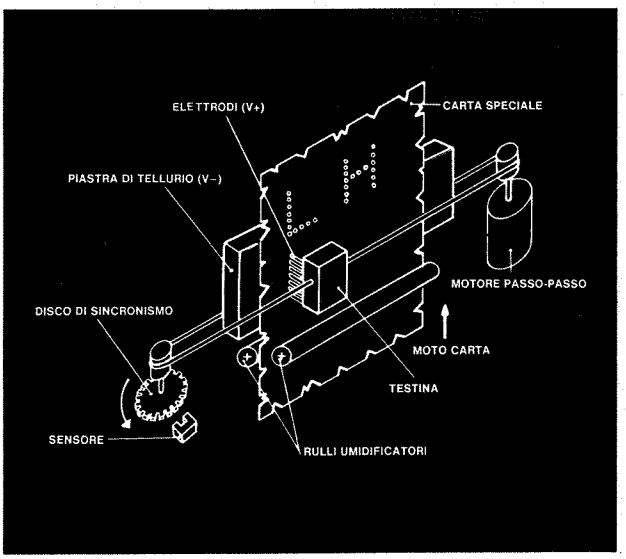


Fig. 12 - Stampante elettrolitica

strato superficiale della carta, scoprendo lo strato nero sottostante.
In alcune macchine la testina è costituita da 16 barrette riscaldanti, ossia il carattere è costruito con una serie di piccoli segmenti anziché con punti. Questo approccio riduce fortemente la velocità (circa 30 car/sec.), per la maggiore inerzia termica della testina.
La stampa termica offre soluzioni costruttive molto semplici e compatte, con poche parti in movimento. Vi sono però alcuni aspetti negativi che ne limitano la diffusione come:

— velocità ridotta (non si superano i 100 car/sec con scrittura a punti ed i 30 car/sec con scrittura a segmenti);
— uso di carte speciali da maneggiare con cura e da utilizzare in ambienti con temperatura ed umidità entro livelli prescritti;
— frequente necessità di pulire i resistori/elettrodi;
— cattivo odore durante la stampa, per la volatilizzazione di sostanze contenute nella carta.

3.8. *Stampa elettrolitica*
In questo sistema (fig. 12) la carta, opportunamente inumidita, per avere buona conduttività, è posta fra la testina di scrittura ed una piastra di tellurio, la quale è tenuta ad un potenziale negativo. Allorchè uno qualsiasi dei sette elettrodi verticali che compongono la testina di scrittura viene portato ad un potenziale positivo, una corrente elettrica fluisce attraverso la carta. Il tellurio viene dissolto elettroliticamente e trasportato dentro la carta, che è di natura porosa. Come risultato si ha un annerimento della carta in corrispondenza della punta dell'elettrodo.
Questa soluzione è semplice e offre buone velocità (attorno ai 250 car/sec.). Per contro, esistono problemi di usura degli elettrodi e l'inconveniente di dover maneggiare carta umida. Queste caratteristiche negative limitano fortemente la diffusione della stampa elettrolitica.

4. Conclusioni

Le stampanti a non impatto sono state oggetto di numerosi studi e progetti negli ultimi dieci anni, ma solo un ristretto numero di modelli è stato costruito in piccole serie per essere utilizzato nei sistemi di elaborazione di dati. Oggi si nota una tendenza a chiarire ed organizzare meglio tutto il settore (tecnico e di vendita) di queste macchine in funzione dell'aumentato interesse da parte degli utenti.
Infatti queste stampanti sono attraenti per le loro caratteristiche funzionali, quali l'elevata velocità, la bassa rumorosità, il livello di affidabilità e inoltre, con gli ultimi sviluppi, stanno diventando interessanti anche per il costo. Questi vantaggi sono in parte attenuati dagli inconvenienti tipici della stampa a non impatto e cioè: uso, in alcuni tipi, di carta speciale; qualità di stampa mediamente inferiore rispetto a quella ottenuta per impatto; impossibilità di avere copie contemporaneamente alla stampa dell'originale. Comunque, in definitiva, è lecito prevedere nei prossimi anni una sostanziale espansione nell'uso di queste stampanti, con una introduzione selettiva nelle applicazioni dove vengono valorizzate le loro caratteristiche peculiari.

5. Bibliografia

D. NICOLAS, *"Les imprimantes hier, aujourd'hui et demain"*, Informatique et gestion, n. 53, 1973.

R. BORELLI, R. BAYLESS, E. TRUAX, *"A nonimpact page printing system"*, Computer, 9/1975.

A. FINOCCHIARO, *"Sistemi elettronici nella stampa e nel plotting"*, Tecnica e automazione, 1971.

"Ultrafast printer merges electrophotographic laser technologies", Computer, 6/1975.

A. RAGOZZINO, M. SHEBANOW, *"Impact and nonimpact printers"*, Honeywell Computer Journal, 1969.

F. FILIPPAZZI, *"Tecnologia dell'eleboratore elettronico: situazione attuale e prospettive"*, edit. F. Angeli, 1976.

Evoluzione dei sistemi di programmazione matematica e delle loro applicazioni

FAUSTO RICCI
Applicazioni Scientifiche
Honeywell Information Systems Italia

1. Introduzione

Con il termine « programmazione matematica » in senso lato si indica generalmente un gruppo di tecniche per la ricerca dei massimi o dei minimi di funzioni di più variabili in presenza di vincoli.
Le funzioni considerate, dette funzioni obiettivo, esprimono ad esempio costi o rischi, da minimizzare, oppure ricavi o profitti, da massimizzare.
I vincoli esprimono limitazioni delle risorse disponibili (materie prime, manodopera), richieste del mercato da soddisfare, capacità degli impianti, ecc.
In questo articolo diamo al termine programmazione matematica un'interpretazione più restrittiva, come è consuetudine parlando di sistemi di software, limitandoci a descrivere sistemi di programmazione lineare (LP = Linear Programming) in cui sia le funzioni obiettivo che le equazioni vincolanti sono lineari, a numeri interi (IP = Integer Programming), in cui le variabili possono assumere solo valori interi, o misti (MIP = Mixed Integer Programming).
Il nostro approccio è motivato anche dal fatto che le altre tecniche di programmazione matematica in senso lato, quale ad esempio la programmazione quadratica e la programmazione stocastica, non sono ancora sufficientemente integrate nei sistemi informativi aziendali. (Per una panoramica abbastanza attuale degli sviluppi di tali tecniche vedi (8) e (9).)
Consideriamo ora un semplice esempio introduttivo.
Un fabbricante, per produrre certi articoli, dispone di due processi produttivi. Con il processo I si possono produrre articoli che chiameremo di tipo α mentre con il processo II si producono articoli di tipo β.
Le risorse necessarie sono:
— manodopera (W) misurata in settimane/uomo
— materia prima (R) misurata in Kg.
— materiale complementare (S) misurato in Kg.
Dato che ogni processo richiede quantità diverse di materiale, come è messo in evidenza dalla Tabella 1, i profitti unitari (in milioni di lire) sono diversi. Il fabbricante deve decidere il livello, che indichiamo con X_1, X_2, di ognuno dei due processi produttivi, in modo da massimizzare il profitto complessivo, tenendo presente che nel periodo considerato egli dispone al massimo di 10 settimane/uomo, 180 Kg. di materia R e 56 Kg. di materiale S.
Le ipotesi di base che permettono la formalizzazione del problema del fabbricante in termini di LP sono due.
1. Proporzionalità: I coefficienti di ogni processo sono relativi a ciascun articolo prodotto mediante quel processo. Ad esempio per produrre 5 articoli di tipo β con il secondo processo supponiamo che siano necessarie 5 settimane/uomo, 100 Kg. di materia prima R e 20 Kg. di materiale S, con un profitto di 40.000 lire. In altri termini l'impegno produttivo ed il profitto sono esattamente proporzionali al numero di articoli prodotti con ogni processo.
2. Additività: Se due o più processi sono utilizzati contemporaneamente, gli impegni produttivi vanno sommati. Ad esempio se vogliamo ottenere 2 articoli con il primo processo e 5 articoli con il secondo, l'impegno complessivo sarà di 2 + 5 = 7 settima-

Tabella 1 - Dati di base del problema del fabbricante (LP)

	Articolo α Processo I	Articolo β Processo II	Disponibilità
Manodopera	1	1	10
Materia prima R	10	20	180
Materiale S	8	4	56
Profitto unitario	9	8	
Livelli di produzione	x_1	x_2	

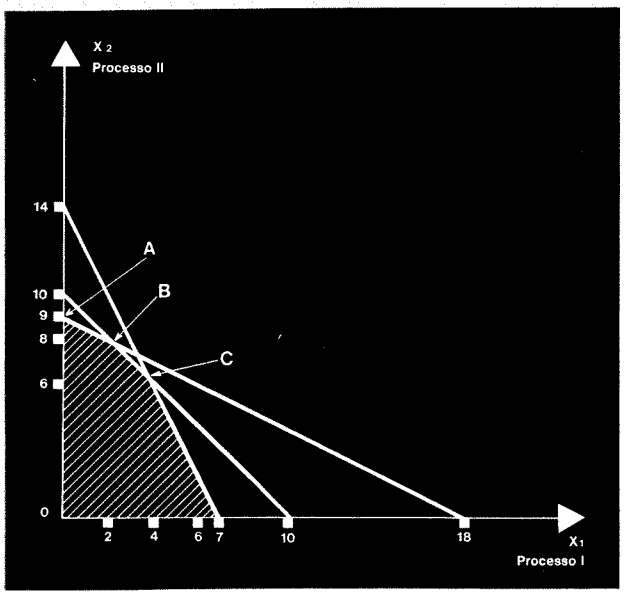


Fig. 1 - Rappresentazione grafica di un problema di programmazione lineare.

ne/uomo, 20 + 100 = 120 Kg di R, 16 + 20 = 36 Kg. di materiale S, con un profitto pari a 18 + 40 = 58 migliaia di lire.
Nel semplice modello che stiamo esaminando non sono quindi permesse economie di scala.

2. Elementi costitutivi di un problema di LP

Utilizzando la terminologia ormai entrata nell'uso comune per i problemi di programmazione matematica, in ogni problema di programmazione lineare sono presenti i seguenti elementi:

— Il vettore C = OBJ (OBJective function = funzione obiettivo) dei coefficienti della funzione da ottimizzare. Nel nostro caso:

$$C = [\,9 \quad 8\,]$$

— La matrice A dei coefficienti produttivi

$$A = \begin{bmatrix} 1 & 1 \\ 10 & 20 \\ 8 & 4 \end{bmatrix}$$

— Il vettore b = RHS (Right Hand Side: vettore dei termini noti) delle disponibilità

$$b = [10 \quad 180 \quad 56]$$

— Il vettore X dei livelli produttivi (incogniti)

$$x = [x_1 \quad x_2]$$

Con la compattezza tipica dell'algebra matriciale un problema di LP può indicarsi con

$$\text{MAX} \quad y = cx$$

$$\begin{cases} Ax \leqq b \\ x \geqq o \end{cases}$$

il che significa che, nel nostro caso, si vuol massimizzare il profitto y rispettando le limitazioni delle risorse ed escludendo, naturalmente, livelli di produzione negativi.
Particolare importanza assumono le dimensioni della matrice A che vengono universalmente indicate con:
— m: N. delle righe (rows)
— n: N. delle colonne (coloumns) o variabili strutturali (structurals)
Nel nostro caso m = 3, in quanto ci sono tre limitazioni fondamentali, ed n = 2 in quanto ci sono due incognite.
Il problema del fabbricante ammette una efficace rappresentazione grafica (Fig. 1) in cui la regione tratteggiata (detta feasible region = regione ammissibile) rappresenta l'insieme delle combinazioni produttive che rispettano i vincoli sulle disponibilità.
Un teorema fondamentale della programmazione lineare dice che la funzione obiettivo raggiunge il suo valore massimo o minimo in corrispondenza di un **vertice** della regione ammissibile (purché questa non sia vuota o non limitata).
Nella Fig. 1 ci sono cinque vertici ammissibili cui si può dare la seguente interpretazione:

0 ($x_1 = 0$; $x_2 = 0$): nessun processo attivato

A ($x_1 = 0$; $x_2 = 9$): solo il processo II attivato

B ($x_1 = 2$; $x_2 = 8$): entrambi i processi attivati

C ($x_1 = 4$; $x_2 = 6$): entrambi i processi attivati

D ($x_1 = 7$; $x_2 = 0$): solo il processo I attivato

Non consideriamo i vertici fuori della regione ammissibile in quanto in essi non sono rispettati i vincoli del problema. Ad esempio la combinazione produttiva ($x_1 = 18$; $x_2 = 0$) viola sia il vincolo sulla manodopera, richiedendo 18 settimane/uomo contro 10, che il vincolo sul materiale S, richiedendone 144 Kg contro 56 disponibili.
I valori della funzione obiettivo, per i vertici ammissibili, sono rispettivamente:

Y (0) = 0 Y (A) = 72 Y (B) = 82
Y (C) = 84 Y (D) = 63

Il profitto massimo si ha nel punto C e si raggiunge:
— attivando il processo I a livello $x_1 = 4$
— attivando il processo II a livello $x_2 = 6$
— utilizzando completamente le 10 settimane/uomo disponibili
— utilizzando 160 Kg di materia prima R sui 180 disponibili
— utilizzando completamente i 56 Kg di materiale S

Abbiamo trovato l'ottimo considerando tutti i vertici ammissibili il che non è evidentemente possibile per i problemi concreti per la cui soluzione si utilizza ancor oggi un procedimento matematico noto come **criterio del simplesso**.
Con tale metodo in una prima fase si determina, se

esiste, un punto della regione ammissibile in cui tutti i vincoli siano rispettati.
Nel nostro caso il vertice suddetto potrebbe semplicemente essere il punto 0 (0; 0).
In una seconda fase si passa da un vertice all'altro della regione ammissibile in modo che il valore della funzione obiettivo non diminuisca (profitti o ricavi) o non aumenti (costi o rischi) nel corso dei vari passaggi chiamati iterazioni.
Con riferimento alla Fig. 1 un tipico cammino potrebbe essere:

(0 con valore Y = 0) → (A con valore Y = 72) →

→ (B con valore Y = 82) → (C con valore Y = 84).

La stessa soluzione potrebbe essere ottenuta con due sole iterazioni seguendo il seguente cammino che nel primo passo (0 → D) sembrerebbe meno conveniente del primo:

(0 con valore Y = 0) → (D con valore Y = 63) →

→ (C con valore Y = 84).

L'enorme vantaggio del criterio del simplesso rispetto a criteri di semplice valutazione dei valori della funzione obiettivo in ogni vertice consiste nella drastica riduzione della mole di calcoli necessari. L'esperienza ha mostrato infatti che il numero di iterazioni necessario per risolvere un problema con n variabili ed m vincoli è in genere compreso fra 1.5 m e 3 m mentre il numero dei vertici che sarebbe necessario esplorare è pari a $\left(\frac{n+m}{m}\right)$.
Ad esempio per un problema di m = 10 vincoli ed n = 10 variabili, qualificabile oggi come piccolissimo, normalmente sono sufficienti da 15 a 30 iterazioni mentre sarebbe necessario valutare il valore della funzione obiettivo in ben $\binom{20}{10} = 184756$ punti.
Problemi medio piccoli con n = m = 500, senza il criterio del simplesso, sarebbero fuori della portata di qualunque elaboratore attuale.

3. Evoluzione storica dei sistemi di programmazione matematica

Il criterio del simplesso è dovuto alla felice intuizione di G.B. Dantzig (1) che può datarsi intorno al 1947. Il primo testo che ci risulti sia stato interamente dedicato alla programmazione lineare è il famoso « An Introduction to Linear Programming », pubblicato nel 1953 (2), familiarmente indicato con « CCH » dalle iniziali degli autori.
Le prime applicazioni riguardano industrie chimiche, dove sono stati impostati e risolti semplici problemi di miscelazione dei derivati del petrolio sin dal 1952. Immediatamente dopo si sono avuti sviluppi nel settore delle industrie manifatturiere (1953) ed alimentari (mangimi).
Mentre gli sviluppi teorici facevano passi da gigante, non altrettanto si poteva dire dei sistemi di calcolo. I primi problemi, risolti con macchine da calcolo tradizionali, non ci risulta superassero 40-50 vincoli ed altrettante variabili (5). Questo deriva dal fatto che la formulazione originale del criterio di Dantzig prevedeva che venissero elaborati tutti i coefficienti di ogni « tavola » del simplesso, composta da m (m + n) elementi.
Un'importante innovazione si ebbe, nel periodo 1954-56, con la messa a punto degli algoritmi PFI (Product Form of the Inverse = inversa sotto forma di prodotto) ad opera di Dantzig e di Orchard-Hays (3), che formano ancor oggi la base dei grossi sistemi di programmazione matematica. Per comprenderne l'essenza bisogna sapere che con questo metodo ogni passo da un vertice all'altro della regione ammissibile (iterazione) può essere compiuto operando soltanto su vettori di m + n elementi invece che su matrici di m (m + n) elementi.
Per problemi più piccoli con m = n = 100 si passa quindi dall'elaborazione di matrici di 10.000 elementi a quella di vettori di 200 elementi, con evidenti vantaggi sia in termini di occupazione di memoria che di tempi di elaborazione. Come spesso accade in Ricerca Operativa, però, gli algoritmi PFI precorrevano i tempi. Se è vero infatti che il programma scritto da Orchard-Hays per elaboratori di grandi dimensioni ammetteva un numero massimo di vincoli pari a 255 (con evidente riferimento al massimo numero rappresentabile con 8 bit) Smith (4) riporta che il tempo di calcolo per un problema con 10 vincoli e 24 variabili, sul 704, un grosso elaboratore a tubi, è stato di 7.2 minuti (oggi si impiegherebbero poche frazioni di secondo). Analoghi risultati si sono avuti verso il 1962 in Italia utilizzando il programma « Simplesso » della serie Elea 9003 che poteva trattare un massimo di 300 vincoli e 2000 variabili, con configurazioni con 6 unità a nastro magnetico, ingresso da scheda o da fotolettore.
Negli anni successivi le dimensioni dei problemi trattabili sono salite rapidamente: citiamo fra tutti il programma Ophelie sviluppato nel 1965 in grado di trattare problemi con 4095 vincoli ed un numero di variabili praticamente illimitato. Per questi problemi un elemento importante è la **densità** della matrice, definita come rapporto fra il numero di coefficienti nonnulli ed il totale degli elementi della matrice A (mn).
I tempi di calcolo su elaboratore della serie 3600 andavano da 5 a 20 minuti per problemi con 500 vincoli e 1000 variabili ed una densità minore dell'1%, a 2-3 ore per problemi con 1000 vincoli e 2000 variabili ed una densità intorno allo 0,5%.
I tempi suddetti sono largamente orientativi in quanto la durata dei calcoli è fortemente influenzata dalla struttura della matrice dei coefficienti e del vettore dei termini noti. Notevole importanza hanno anche i parametri dello stesso programma di calcolo, che si possono paragonare alle viti micrometriche dei carburatori di una delicata macchina da corsa. Con opportune strategie, che appartengono al dominio

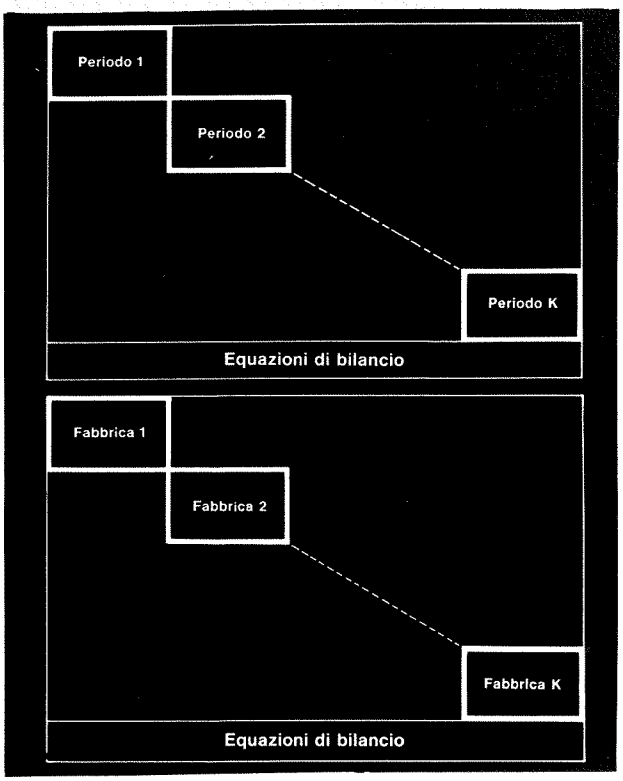


Fig. 2 - Tipiche strutture della matrice dei coefficienti nei problemi di L.P.

dell'analisi numerica sofisticata, è possibile ad esempio raggiungere l'ottimo con una lunga successione di passi « brevi » oppure con un numero relativamente più basso di passi « lunghi », controllando la precisione e la velocità dei calcoli (con le cosiddette reinversioni) spesso oppure raramente, e così via (anche nel semplice esempio della fig. 1 abbiamo visto che sono possibili due diverse strategie per raggiungere l'ottimo). Nel periodo dal 1960 al 1965 la programmazione matematica ha compiuto un salto qualitativo notevole, principalmente per merito di R. Gomory e E. Balas, con l'implementazione di algoritmi di programmazione discreta o mista, in cui tutte le variabili o una parte di esse è suscettibili di assumere soltanto valori interi non negativi.

Questa limitazione complica enormemente la soluzione dei problemi di ottimizzazione (essenzialmente perché la regione ammissibile non è più convessa), tanto che Haldi (5) nel 1965 riporta che i problemi risolubili in un tempo ragionevole arrivavano al massimo a 50 vincoli e 50 variabili intere.

Dal 1966 al 1970 le possibilità di soluzione sono aumentate, sia per un certo miglioramento degli algoritmi ma soprattutto per l'aumento delle velocità degli elaboratori e delle dimensioni di memoria disponibili.

Sino alla fine degli anni '60 però, salvo rare eccezioni, i sistemi di programmazione matematica sono rimasti puri strumenti di calcolo, ancorché poderosi, senza essere integrati nei sistemi informativi aziendali.

È soltanto a partire dal 1970 che si sono fatti dei progressi in questa direzione con la messa a punto di linguaggi generatori di matrici e di formati di stampa.

Per capirne l'importanza notiamo come il calcolo delle soluzioni di un problema di programmazione lineare sia soltanto un passo del lungo cammino che è necessario compiere per realizzare un modello e verificarne l'efficacia.

In una prima fase è necessario procedere alla formalizzazione generale del modello, aiutati in questo da una consistente serie di esempi disponibile nella letteratura tecnica. In una seconda fase è necessario organizzare la matrice dei coefficienti tecnici che molto spesso ha una caratteristica struttura a blocchi (vedi fig. 2).

I singoli blocchi di coefficienti possono riferirsi a diversi periodi di tempo (programmazione temporale o time-phasing) o a diverse unità produttive (programmazione spaziale), raramente ad entrambe le situazioni. Ci sono poi equazioni di bilancio che legano i blocchi fra loro. Per la generazione delle matrici sono stati sviluppati linguaggi chiamati MGL (Matrix Generation Languages) composti da potenti verbi per il trattamento di informazioni alfanumeriche. Ad esempio, data la matrice di costi di trasporto seguente

$$T = \begin{array}{c|ccc} & MI & TO & NA \\ GE & C_{11} & C_{12} & C_{13} \\ RO & C_{21} & C_{22} & C_{23} \end{array}$$

mediante la frase A = TRANSP (T, X7, COSTO) viene creata la matrice

$A =$

	MI: GE: X7	MI: RO: X7	TO: GE: X7	TO: RO: X7	NA: GE: X7	NA: RO: X7
MI	1	1				
TO			1	1		
NA					1	1
GE	-1		-1		-1	
RO				-1		-1
COSTO	C_{11}	C_{21}	C_{12}	C_{22}	C_{13}	C_{23}

Notiamo che con questa frase non soltanto viene creato l'intero gruppo di coefficienti per un problema di trasporti da tre origini (MI, TO, NA) a due destinazioni (GE, RO), ma che vengono anche creati i nomi delle variabili corrispondenti.

Ad esempio MI:RO:X7 corrisponde alla quantità di prodotto X da trasportare da Milano a Rovigo nel 1977, e così via.

Una volta generata la matrice A viene chiamato il programma di calcolo mediante verbi che fanno parte del cosiddetto ACL (Agenda Control Language = linguaggio di controllo e soluzione) che permettono di trovare, se esiste, la soluzione ottimale che corrisponde al minimo costo o rischio oppure al massimo profitto.

In questo ambito a volte è conveniente utilizzare metodi di soluzione che scompongono la matrice in blocchi: si tratta degli algoritmi di decomposizione realizzati dal 65 al 75, che permettono di risolvere problemi anche di 20.000 vincoli e 200.000 variabili. Problemi con 15.000 vincoli e 50.000 variabili, con una densità del 2-3% sono oggi risolti su elaboratori di grandi dimensioni in poche ore di calcolo.

Una volta ottenuta la soluzione spesso se ne vuole

provare la « sensitività » a cambiamenti nei coefficienti. Si parla in questo caso di « Ranging » (o analisi marginale), quando varia un solo coefficiente per volta, ad esempio la disponibilità di una certa materia prima. Si parla invece di « Parametrizzazione » quando varia un intero gruppo di coefficienti, ad esempio quando aumentano tutti i costi contemporaneamente.
Naturalmente l'analisi delle prime soluzioni spesso suggerisce modifiche al modello di programmazione lineare, realizzabili abbastanza facilmente mediante verbi (Revise e Modify) che cambiano i valori dei coefficienti della matrice **A**, del vettore **b** dei termini noti o del vettore **c** dei costi/ricavi.
Arrivati ad una soluzione accettabile subentra l'ultimo passo della realizzazione di un modello di LP cioè la stampa dei risultati.
L'output standard dei sistemi di programmazione matematica è in genere comprensibile solo agli specialisti del ramo. Per ovviare a questo inconveniente sono stati realizzati linguaggi generatori di formati di stampa (F.G.L. = Format Generator Language) mediante i quali è possibile ottenere tabulati o estremamente dettagliati, con esplosione delle informazioni elementari, che vanno ai responsabili operativi delle varie attività, o estremamente concisi, per il top management, dove vengono messe in evidenza (in genere dal punto di vista economico-finanziario) le conseguenze derivanti dall'adozione di strategie specifiche.
Dal '75 in poi infine è stato fatto uno sforzo per inserire ancora di più nei sistemi informativi i sistemi di programmazione matematica, che hanno nomi auto-esplicativi come gli MPS (Mathematical Programming System) o di fantasia come Apex, Ilona, Umpire.
Sono nati veri e propri linguaggi di gestione di un data base scientifico, come il CFMS (Common File Management System), che costituiscono un valido, indispensabile aiuto per i ricercatori operativi che si dedicano alla analisi ed alla realizzazione di grossi modelli.

4. Applicazioni

Possiamo distinguere varie classi di applicazioni delle tecniche di programmazione matematica.
Ad una prima classe appartengono modelli di piccole dimensioni (20-50 vincoli e 20-100 variabili) per risolvere problemi di miscugli.
L'esempio più classico è quello della minimizzazione del costo delle formule di mangimi per animali. Le variabili sono le materie prime disponibili sul mercato mentre i vincoli corrispondono alle percentuali minime e massime dei vari elementi nutritivi (vitamine, proteine, calorie) che devono essere presenti nei mangimi.
Sistemi di LP in grado di risolvere tali problemi in pochi minuti sono disponibili anche su elaboratori di piccole dimensioni.
All'utente in genere non è richiesta alcuna cognizione matematica: è sufficiente infatti fornire i dati di

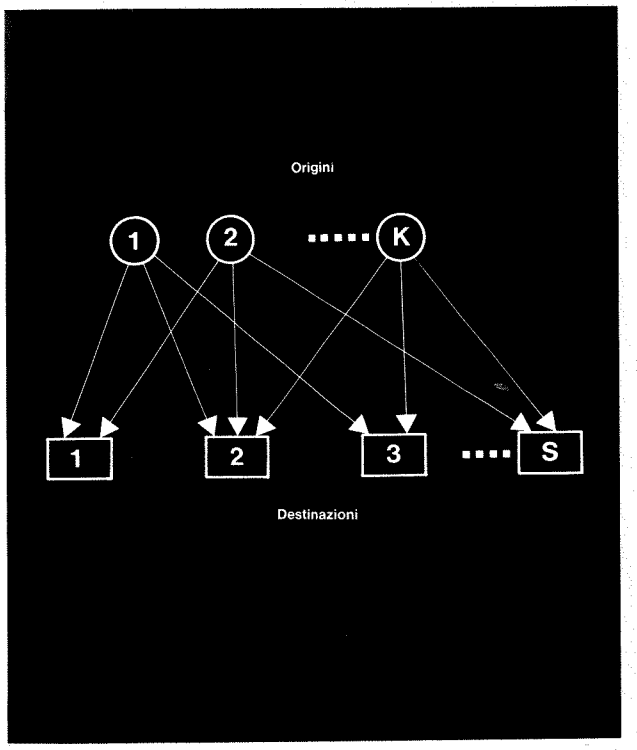


Fig. 3 - Problemi di trasporto

analisi e di costo per avere le formulazioni praticamente immediatamente.
Problemi matematici molto simili a quello dei mangimi sono quelli di formazione di leghe partendo da differenti partite di materiale.
L'applicazione è più complessa, come è stato messo in evidenza nell'articolo « la programmazione lineare non sempre è la risposta » (7), che suggerisce molti motivi di riflessione ai ricercatori operativi.
Viene infatti correttamente posto in evidenza come la risposta ottenibile applicando metodi di LP in questo caso non sia molto affidabile, in quanto l'analisi delle partite di rottami è spesso molto approssimativa. D'altronde, viene affermato, il costo di un'analisi accurata supera il vantaggio ottenibile con tecniche di programmazione matematica, a meno che non si superi il milione di tonnellate l'anno di materiale fuso.
Ad una seconda classe appartengono problemi di trasporto. Data una serie di K origini in cui è disponibile un certo prodotto (ad esempio fabbriche o magazzini) ed una serie di S destinazioni (negozi o centri di distribuzione) e data una matrice di costi di trasporto, da ogni origine ad ogni destinazione, il problema dei trasporti consiste nel rendere minimo il costo complessivo di trasporto rispettando i vincoli di disponibilità e soddisfacendo le richieste (vedi fig. 3).
Se i costi sono proporzionali alle quantità trasportate il problema dei trasporti rientra fra quelli di programmazione lineare con $n = KS$ variabili ed $m = K + S$ vincoli.
Stante la natura particolare del problema questo può venir risolto con algoritmi particolari, richiamati dal verbo TRANSP, molto più veloci dell'algoritmo del simplesso.

Problemi con 100 origini e 100 destinazioni, cioè con 200 vincoli e 10.000 variabili, sono oggi risolvibili in pochi minuti di elaborazione.
Lo sforzo richiesto all'utente è principalmente quello della rilevazione dei costi di trasporto che normalmente dipendono dalle distanze fra le origini e le destinazioni.
Fortunatamente tale sforzo è compiuto una sola volta ed è anche per questo che problemi di trasporto concreti sono risolti abbastanza spesso dalle aziende.
Non bisogna confondere i problemi di trasporto con i problemi di itinerari, in cui occorre scegliere l'ordine con cui toccare una serie di centri di distribuzione per minimizzare la distanza complessiva percorsa. Si tratta dei cosidetti problemi del commesso viaggiatore (T.S.P. = Travelling Salesman Problem) che ancor oggi non sono normalmente risolti in modo esatto bensì con algoritmi euristici che escono dal dominio dei sistemi standard di programmazione matematica.
Ad una terza classe appartengono i problemi cosidetti « dello zaino » (Knapsack) in quanto derivano dal semplice problema di massimizzare l'utilità complessiva di un certo numero di oggetti, di volume dato, da introdurre in un zaino di determinata capacità. Matematicamente si tratta di problemi di programmazione discreta, in quanto non è accettabile considerare un numero frazionario di oggetti, soprattutto nel caso frequente di variabili booleane che possono assumere soltanto i valori 0 (nessun oggetto) o 1 (1 oggetto). Come già accennato i problemi di programmazione a numeri interi sono di difficile soluzione ancor oggi appena il numero delle variabili supera il centinaio; nel caso dei problemi dello zaino però gli algoritmi sono abbastanza efficienti anche per problemi di grandi dimensioni. Tipiche applicazioni si hanno per il problema degli sfridi che si ha per esempio nelle cartiere o nelle aziende che tagliano nastri metallici. Si tratta di ricavare, da un nastro di grandi dimensioni, nastri di dimensioni ridotte per far fronte ad un carnet di ordini in modo da minimizzare lo spreco di materiale chiamato sfrido (vedi fig. 4a).
Alla stessa classe appartengono i problemi che nascono quando si vogliono ricavare da formati rettangolari di grandi dimensioni rettangolari di dimensioni minori (vedi fig. 4b), come avviene ad esempio nelle aziende che tagliano riquadri per elettrodomestici. L'utilità di programmi che minimizzano gli sfridi, attualmente abbastanza diffusi a livello sperimentale, potrà crescere nel prossimo futuro con una maggiore integrazione degli stessi nei sistemi di programmazione della produzione.
Ad una quarta classe appartengono modelli di scelta fra più investimenti alternativi, con l'obiettivo di massimizzare i rendimenti o minimizzare i rischi rispettando vincoli di composizione, ad esempio del portafoglio titoli. Per quanto nella letteratura si trovino molti modelli di questo tipo riteniamo che essi non riescano, se non in rari casi, a superare la fase didattica per entrare in quella ben più difficile dell'applicazione concreta e sistematica. Maggiori

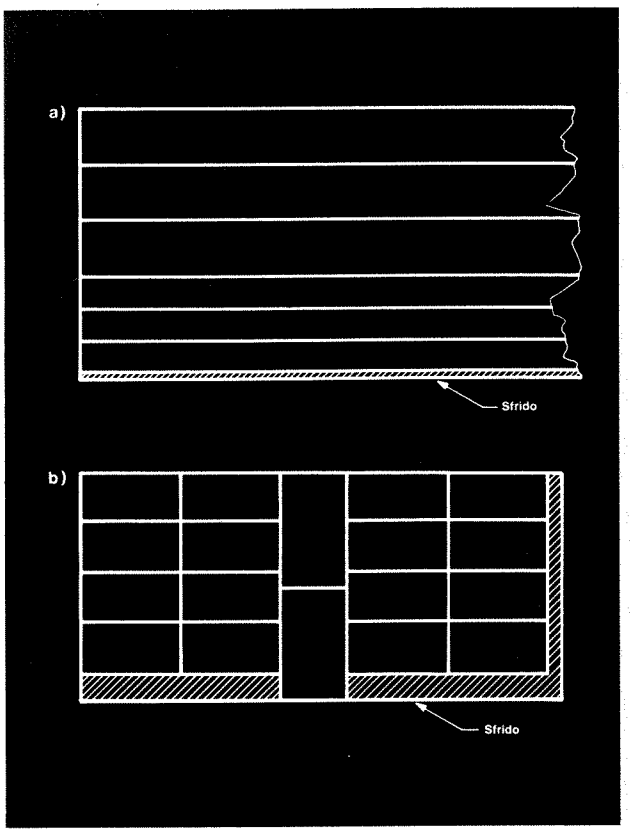


Fig. 4 - Problemi di minimizzazione degli sfridi

possibilità di sviluppo hanno invece in questo ambito modelli di simulazione o semplici modelli di « WHAT IF » (Cosa succede se).
Ad un'ultima, estesa classe appartengono i modelli di gestione aziendale le cui applicazioni hanno proceduto di pari passo con l'evoluzione degli elaboratori. Si è iniziato all'incirca nel 1960 a sviluppare alcuni piccoli modelli di programmazione lineare nell'industria petrolifera, con poche decine di vincoli e limitando l'analisi a pochi impianti fondamentali. In questa fase molto importante sotto l'aspetto didattico e di formazione di specialisti si sono anche commessi errori di prospettiva nel sopravalutare all'inizio sia le tecniche che le possibilità di elaborazione. Se è vero che il concetto di modello di LP è apparso subito molto efficace per descrivere alcuni tipici processi di produzione, in sede pratica si sono avute molte difficoltà in quanto automatizzare il calcolo di un programma di produzione inserendo i dati di una « black box » suscitava sospetti e sfiducia. Inoltre i primi risultati erano molto approssimati per l'inadeguatezza dei sistemi di rilevazione e spesso di scarsa utilità.
La situazione è molto migliorata dal '62 al '65 con l'introduzione nelle aziende di procedure contabili standard che, utilizzando compiutamente gli elaboratori, erano in grado di fornire una massa di dati analitici di base per i modelli di programmazione matematica, specialmente nel caso già citato delle aziende petrolchimiche.
Lo sviluppo degli algoritmi di programmazione mista

ha esteso notevolmente la possibilità di applicazione di tali modelli. Citimo ad esempio i grossi modelli realizzati in Italia per lo sviluppo dei grandi porti petroliferi (1970), che comprendevano circa 56000 variabili continue, 240 variabili intere e 2600 vincoli ed hanno fornito una serie di risultati utili sia a livello operativo immediato che a livello strategico.

Ancora citiamo i modelli congiunti di produzione e trasporto di un grossa azienda americana che opera nel settore dei derivati del legno e che ottimizza i profitti per periodi dell'ordine di 10 anni, con modelli di 15000 equazioni e 50000 variabili, arrivando persino a pianificare la crescita di piante per i successivi 20 anni!

Come si vede dall'esposizione dei casi, ovviamente limitata a quelli che ci sono sembrati più interessanti, esiste, in Italia, una sufficiente serie di esperienze, sia in termini di formulazione dei modelli che di realizzazione, cui riallacciarsi per risolvere concretamente anche in aziende di piccole dimensioni, problemi di programmazione matematica, chiedendo agli elaboratori quell'ausilio che essi possono e debbono dare, e ancora di più daranno in un prossimo futuro.

5. Bibliografia

[1] Activity Analysis of Production and Allocation (T.C. Koopmans, Ed.), G. B. Dantzig, *"Maximization of a linear function of variables subject to linear inequalities"*, Chap. XXI, pp. 339-347, Wiley, New York, 1951.

[2] A. CHARNES, W. W. COOPER, A. HENDERSON, *"An introduction to linear programming"*, Wiley, New York, 1953.

[3] W. ORCHARD-HAYS, *"Evolution of computer codes for linear programming"* P-810, The rand Corporation, March 1956.

[4] S. B. SMITH *"Planning transistor production by linear programming"*, J.O.R.S.A. 1965, p. 132.

[5] H. M. WAGNER, *"A two-phase method for the simplex tableau"*, J.O.R.S.A. 1956, p. 443.

[6] J. HALDI, L. M. JSAACSON *"A computer code for integer solution to linear programs"* J.O.R.S.A., 1956, p. 946.

[7] D. E. DEBEAM, *"Linear programming isn't always the answer"*, J.O.R.S.A. 1957, p. 429.

[8] D. F. SHANNO, R. L. WEIL, *"Management science: a view from nonlinear programming"*, Comm. A.C.M. 1972, p. 542.

[9] I. M. STANCU-MINASIAN, M. J. WETS, *"A research bibliography in stochastic programming 1955-1975"*, J.O.R.S.A. 1976 p. 1078.

La progettazione dei sistemi informativi aziendali

PASQUALE BLASIO
Sezione Formazione
Honeywell Information systems Italia

1. Introduzione

Volendo tracciare una sintesi della evoluzione che negli ultimi 20 anni ha caratterizzato l'utilizzazione dell'informatica nell'ambito aziendale, si possono individuare tre periodi.

Il primo periodo che grossolanamente coincide con la 2ª generazione di sistemi EDP (1955-1964) è caratterizzato dalla proliferazione nella azienda di procedure meccanizzate di carattere settoriale concentrate specialmente nell'area amministrativo-contabile.

Gli archivi che raccolgono e organizzano i dati di queste procedure sono generalmente di tipo sequenziale e non sono tra loro intercomunicanti. Con il risultato che, ad esempio le informazioni comuni alle funzioni contabilità clienti e fatturazione compaiono duplicate sui due archivi relativi.

Il tipo di competenza richiesto al personale addetto al sistema EDP è di natura tecnica sostanzialmente orientato a padroneggiarne l'impiego nello sviluppo e nella esecuzione delle procedure le cui specifiche vengono definite sulla base delle esigenze esplicitate dalle varie funzioni aziendali.

L'inserimento del settore automazione è nel dipartimento aziendale che ne è il principale utilizzatore, tipicamente l'amministrazione.

Non esistono in questo periodo modalità formalizzate di controllo dei progetti e delle conseguenti spese di informatica anche perché la principale giustificazione all'automazione è la riduzione o quanto meno il contenimento delle risorse addette a lavori di carattere operativo.

Il secondo periodo che, molto orientativamente si estende dal 1964 alla fine degli anni 70, è caratterizzato da un certo numero di fatti tra loro correlati e che si presentano contemporaneamente all'utenza EDP.

Viene introdotta la 3ª generazione di sistemi di elaborazione che, con Il suo favorevole rapporto prezzo/prestazioni e con le sue superiori capacità operative rende rapidamente obsoleto il parco macchine istallate e stimola l'utenza ad una evoluzione tecnologica.

Ci si rende però ben presto conto che l'impiego efficiente di queste nuove possibilità è condizionato da un diverso modo di progettare il sistema informativo.

L'esigenza della transizione alla 3ª generazione avviene contemporaneamente alla presa di coscenza che, non appena si voglia, dalle attività operative estendere la automazione alle funzioni di controllo gestionale che necessariamente coinvolgono più settori aziendali, tutto l'insieme di procedure settoriali sviluppate nel periodo precedente va sostanzialmente riconcepito se si vogliono eliminare le incongruenze e si vuole garantire un livello accettabile di adattabilità e di prestazioni.

Il rapporto tra i responsabili EDP e i dirigenti dei settori operativi aziendali tende a divenire in questo periodo un aspetto critico della efficienza di impiego del sistema informativo aziendale.

Da una parte l'inserimento del gruppo EDP nell'Amministrazione tende a privilegiare le esigenze applicative di questo settore che di conseguenza diviene anche quello più preparato a valersene, dall'altro la differenza di formazione professionale fra i tecnici di informatica e i responsabili aziendali a livello operativo rende obiettivamente problematico il dialogo con il risultato che è in questo periodo che si verificano quei grossolani errori di valutazione nell'impegno dei vari progetti che hanno alimentato per anni la pubblicistica sull'argomento.

Per citare K. Grindley e J. Humble nel loro « The Effective Computer » è questo il periodo in cui gli uomini EDP giudicano « arretrata » la mentalità dei dirigenti operativi e a loro volta sono da questi ultimi giudicati privi di sensibilità ai problemi concreti dell'azienda.

È caratteristica di questo periodo una esplosiva crescita dei costi EDP (sia in termini di apparecchiature che di risorse) che non trova corrispondenza in un confrontabile aumento della efficacia aziendale dell'informatica.

Si sviluppano di conseguenza procedure di controllo delle spese sovente con obiettivi di dichiarato ridimensionamento e senza tener conto della pecularietà degli investimenti in informatica.
La qualificazione del personale operante nel settore EDP è sostanzialmente inalterata salvo una forte spinta ad acquisire maggiore sensibilità nei problemi di pianificazione dei progetti. Cominciano ad imporsi a livello delle aziende di maggiori dimensioni standard basati su modelli del processo di sviluppo « life cycle ».
Dal punto di vista degli strumenti di analisi è alla fine di questo periodo che per venire incontro alla sempre più diffusa esigenza di un approccio più « scientifico » si rendono disponibili metodologie di progettazione dei sistemi informativi volte a razionalizzare il ruolo dell'informatica attraverso una visione globale delle relazioni funzionali nell'interno dell'azienda e nei rapporti con il mondo esterno.
Di queste nuove concezioni, l'esempio forse più completo e organico è costituito dal BISAD (*) che viene appunto introdotto in questo periodo e successivamente perfezionato per arrivare alla attuale versione.
Nel 3° periodo che copre gli anni 70 ci si rende conto che la chiave del successo di un sistema informativo dipende sia dal coinvolgimento della Direzione Aziendale nella definizione dei suoi obiettivi, sia dalla capacità degli analisti di comprendere effettivamente le problematiche dell'azienda nel suo complesso e quelli dei suoi vari settori operativi per integrarle in una visione unitaria.
Si tende cioè a vedere il sistema informativo non più come produttore di elaborazioni specifiche ma come supporto essenziale al sistema di comunicazioni aziendali che deve essere alimentato con continuità ai suoi vari livelli di sintesi.
La conseguenza di questo approccio è la fusione o cooperazione che si viene istituendo tra le funzioni organizzazione e informatica con il conseguente sviluppo della figura di analista di sistema informativo cui propriamente si riferiscono le metodologie di impostazione e controllo dei sistemi informativi già introdotte alla fine del precedente periodo ma che solo ora trovano il ruolo aziendale capace di recepirle.
Dal punto di vista della tecnologia EDP, a questa evoluzione dell'ottica aziendale nei riguardi del sistema informativo corrisponde lo sviluppo dei sistemi interattivi (terminali conversazionali collegati su linea al sistema di elaborazione) gravitanti su basi di dati integrate contenti la totalità delle informazioni necessarie allo svolgimento delle varie funzioni.
Prima di procedere ad una breve esposizione delle caratteristiche del BISAD è interessante notare come la precedente articolazione in tre periodi della evoluzione storica nell'impiego dell'informatica, corrisponda in una visione sincronica, ai diversi livelli di meccanizzazione riscontrabili ancora oggi nelle varie fasce della utenza EDP.

2. Il BISAD

Come accennato, il BISAD è il metodo suggerito dalla Honeywell per la progettazione e messa a punto del sistema informativo aziendale, e rappresenta un approccio per la soluzione dei problemi che nascono prima, durante e dopo il suo studio e la sua realizzazione.
La sigla BISAD sta a precisare che si procede al disegno di un sistema informativo con una filosofia « business oriented » nel senso che almeno inizialmente non si tiene in considerazione l'eventuale elaboratore.
Il BISAD fornisce un metodo scientifico per analizzare il sistema azienda e progettarne i flussi informativi, colmando quella lacuna di comunicazioni tra analisti e Direzione con il coinvolgimento di tutte le componenti aziendali e la verifica costante della rispondenza del progetto alle direttive del management aziendale. Il BISAD è un metodo che consente di percorrere passo passo la sequenza delle attività aziendali da compiere. Esso permette di evitare di prendere in esame le procedure esistenti e quindi di rimanere ancorati a soluzioni obsolete, ma punta direttamente allo sviluppo di un sistema globale concentrando tutte le risorse verso quell'obiettivo senza segmentarle e disperderle in applicazioni settoriali spesso addiritura casuali. Ignorando poi i limiti posti dall'organizzazione gerarchica, il metodo impone di sviluppare il progetto dei flussi informativi in modo da soddisfare le esigenze che nascono dal fatto che l'azienda è articolata in varie funzioni tutte miranti a raggiungere gli obiettivi aziendali. A grandi linee si può dire che con il BISAD ci si propone di:
1. definire le attività ed i compiti svolti nella azienda per poter conseguire gli obiettivi e raggrupparle in settori logici, cioè in « Funzioni » che nulla hanno a che fare con i reparti o le Direzioni in cui l'azienda può essere organizzata;
2. mettere in evidenza le correlazioni informative ossia il legame esistente, sotto forma di flussi di informazioni, tra tutte le attività;
3. disegnare il modello prototipo del sistema informativo aziendale che mostri già in termini operativi, anche se non necessariamente ad uso della elaborazione elttronica, i dati in ingresso e quelli in uscita delle singole attività e le informazioni che si devono impiegare perché si passi dai primi ai secondi;
4. definire i metodi ed i tempi di realizzazione del sistema progettato.

Per il raggiungimento degli obiettivi stabiliti il BISAD propone un'organizzazione del lavoro secondo i seguenti passi:
1. Analisi di Base
2. Analisi Funzionale
3. Disegno del Prototipo
4. Specifiche del Sistema Informativo

* Business Information Systems Analysis and Desig.

3. L'analisi di base

Chiunque svolga una qualsiasi attività in una azienda, diventa familiare con le problematiche del settore in cui opera. Per esempio il progettista si interessa dei metodi di produzione dell'azienda, il venditore vede l'azienda sotto l'aspetto Marketing e così via. Ciascuna di queste persone vede l'azienda in funzione della propria specializzazione e nessuna di esse ha necessariamente una visione complessiva di tutta l'azienda. Così il primo obiettivo dell'analista è quello di individuare e definire i problemi dell'azienda e quindi impostare e sviluppare una soluzione per quei problemi. L'analista deve trattare ciascuno e tutti gli aspetti della realtà aziendale e per fare ciò egli deve disporre di chiare e precise informazioni. Egli deve porsi in grado di raccogliere queste informazioni in un ragionevole lasso di tempo e deve sapere come valutarle ed organizzarle.

L'analisi di base costituisce la piattaforma delle esigenze aziendali in termini di informazioni. In questa fase si utilizzano tutte le tecniche note per raccogliere le informazioni sull'azienda: inerviste, indagini ed osservazioni dirette, letture di documenti, emissione di questionari e via di seguito. La migliore fonte di informazione è indubbiamente rappresentata dal management della stessa azienda e l'intervista è la tecnica più efficace per ottenere informazioni. Generalmente le interviste vengono condotte dall'analista con tre gruppi di persone:

1. Management decisionale
2. Management operativo
3. Personale operativo

Le interviste col management decisionale devono mirare ad ottenere informazioni su politiche ed obiettivi aziendali ed inoltre sulla storia e sviluppo dell'azienda. Le interviste con il management operativo devono essere orientate sul funzionamento dei vari sottosistemi aziendali di cui sono responsabili, sui rapporti con gli altri settori dell'azienda. Le interviste a questo livello dovranno cioè coprire le seguenti aree:

1. Procedure di lavoro
2. Rapporti con gli altri settori aziendali
3. Problemi attuali
4. Sviluppi futuri

Infine le interviste con il personale operativo caratterizzate dalla raccolta di informazioni di dettaglio devono riguardare la loro specifica attività nell'ambito del contesto del lavoro globale nel proprio settore. Al fine di poter utilizzare in modo razionale le informazioni raccolte è necessario organizzarle e classificarle e per soddisfare le esigenze del BISAD esse devono essere organizzate secondo i seguenti punti:

1. Storia e sviluppo del settore di mercato e dell'azienda
2. Consuetudine del settore
3. Leggi e regolamenti
4. Goals ed obiettivi aziendali
5. Organizzazione dell'azienda
6. Risorse (uomini, mezzi, macchine, materiali)
7. Prodotti e/o servizi offerti
8. Caratteristiche operative
9. Caratteristiche della concorrenza
10. Livello di esperienza in EDP
11. Problemi generali

4. L'analisi funzionale

L'analisi di base ha consentito la raccolta delle informazioni riguardanti il settore industriale in generale e l'Azienda in particolare, essa ha consentito cioè di creare la « big picture » dell'azienda, ed è soltanto sfruttando questo lavoro che l'analista può iniziare il suo studio sulle esigenze del Sistema Informativo. Bisogna ora procedere a dividere la « big picture » in aree logiche, a studiare ciascuna area, ed a ricercare correlazioni informative che intercorono tra le diverse aree logiche. Nella terminologia BISAD ciascuna area logica prende il nome di Funzione ed è rappresentata da un gruppo di attività omogenee.

Il BISAD suggerisce di sviluppare l'analisi funzionale secondo i seguenti passi.

a. Sviluppare la Carta delle Ativită
b. Costruire l'Interfaccia totale delle informazioni.
c. Costruire le Interfacce funzionali delle informazioni.

a) Sviluppare la Carta delle Attività: si chiama Carta delle Attività una lista riportante tutte le attività riscontrate durante l'analisi di base, essa si basa su un concetto fondamentale della problematica aziendale: qualsiasi azienda è sempre composta da una serie finita di attività.

Queste attività si devono raggruppare in aree logiche chiamate appunto Funzioni. La figura 1 riporta un esempio di Carta delle Attività.

È importante ricordare che l'analista dovrà preoccuparsi soltanto di « che cosa » è il contenuto delle attività e non di « chi » svolge l'attività, « dove » si svolge e « come » si svolge.

b) Costruire l'Interfaccia totale delle informazioni: l'obiettivo dell'interfaccia totale è quello di evidenziare i flussi informativi che intercorrono tra le diverse Funzioni previste sulla Carta delle Attività.

La tecnica suggerita è quella di considerare una Funzione per volta e ricercare tutte quelle informazioni fondamentali che consentono lo svolgimento di tutte le attività previste nell'ambito della Funzione, ignorando origine e destinazione rispettivamente dei flussi informativi in input e di quelli output e successivamente legare tutte le Funzioni attraverso flussi informativi creando così la Interfaccia totale delle informazioni.

È evidente che tutti i flussi individuati e convalidati saranno descritti indicando a fronte di ciascuno il contenuto, il motivo, la frequenza, il volume, la latenza etc.

Nella figura 2 è riportato un esempio di Interfaccia totale, dove i rettangoli rappresentano le Funzioni, le linee continue i flussi interni e quelle tratteggiate i flussi di interfaccia con l'esterno dell'azienda.

c) Costruire le interfacce funzionali delle informazioni: dopo aver documentato l'aspetto globale dell'a-

BISAD — **CARTA DELLE ATTIVITÀ**

FUNZIONI	ATTIVITÀ 1	2	3	4	5	6	7	8
A DIREZIONE E PIANIFICAZIONE	Controllo Fattori Esterni e Rapporto agli Azionisti	Controllo Operazioni ed Emissioni Politiche	Emissione Controllo del Budget	Emissione e Controllo degli Standards				
B CONTROLLO AMMINISTRATIVO	Profitti e Perdite	Budget	Contabilità Vendite	Contabilità Acquisti	Contabilità Costi			
C MARKETING	Conciliazione tra Mercato e Prodotti Potenziali	Previsione Domanda	Promozione Vendite	Ricevimento Ordini	Indagine su Reclami e Suggerimenti dei Clienti			
D CONTROLLO STOCK	Determinazione Richieste di Approvvigionamento	Ricevimento e Controllo Merce	Immagazzinamento Merci ed evasioni ordini	Programmazione e Gestione Consegne a clienti				
E PRODUZIONE	Determinazione Specifiche di Produzione	Ricerca e Sviluppo Nuovi Prodotti	Programmazione della Produzione	Gestione e Controllo della Produzione	Gestione e Manutenzione Impianti			

Fig. 1 - Esempio di carta delle attività

zienda tramite l'interfaccia totale, il successivo passo logico è quello di considerare ed analizzare con maggiore dettaglio ciascuna Funzione.
Ciò si realizza evidenziando le realzioni tra attività e flussi informativi nell'ambito di ciascuna Funzione e con le altre Funzioni collegate.
Inoltre per ciascuna attività si dovrà indicare la frequenza con cui l'attività utilizza le informazioni di cui ha bisogno e genera le informazioni da fornire.
Questa rappresentazione si chiama Interfaccia funzionale ed è chiaro che vi saranno tante interfacce quante sono le Funzioni previste nella Carta delle Attività.
La scomposizione dell'Azienda in Funzioni ed Attività facilita notevolmente il lavoro dell'analista ed inoltre il management è in grado di comprendere simboli e terminologia usata senza rimanere confuso da un linguaggio tecnico incomprensibile.
Nella figura 3 è riportato un esempio di Interfaccia funzionale dove i rettangoli a linea continua indicano le attività e quelli tratteggiati le Funzioni collegate.

5. Disegno del prototipo

Finora l'analista si è preoccupato di individuare quali sono le attività che si svolgono nell'ambito dell'azienda e quali sono i flussi informativi necessari.
Si tratta ora di trasformare le singole interfacce funzionali in prototipi funzionali considerando « come » le attività si devono svolgere e trasformando i flussi informativi in informazioni classificandoli in due gruppi:

- Veicoli di dati
- Centri di informazione

Per Veicoli di dati si intendono tutti i rapporti manageriali, documenti operativi, documenti di controllo ed in generale tutte le informazioni che si trattano una sola volta.
Per Centri di informazioni si intendono invece gli insiemi logici ed omogenei di elementi.
Un Centro di informazioni viene usato più volte, può essere aggiornato o rinnovato e soprattutto può essere utilizzato da una o più attività.
La simbologia da impiegare per il disegno dei Prototipi funzionali è la seguente:

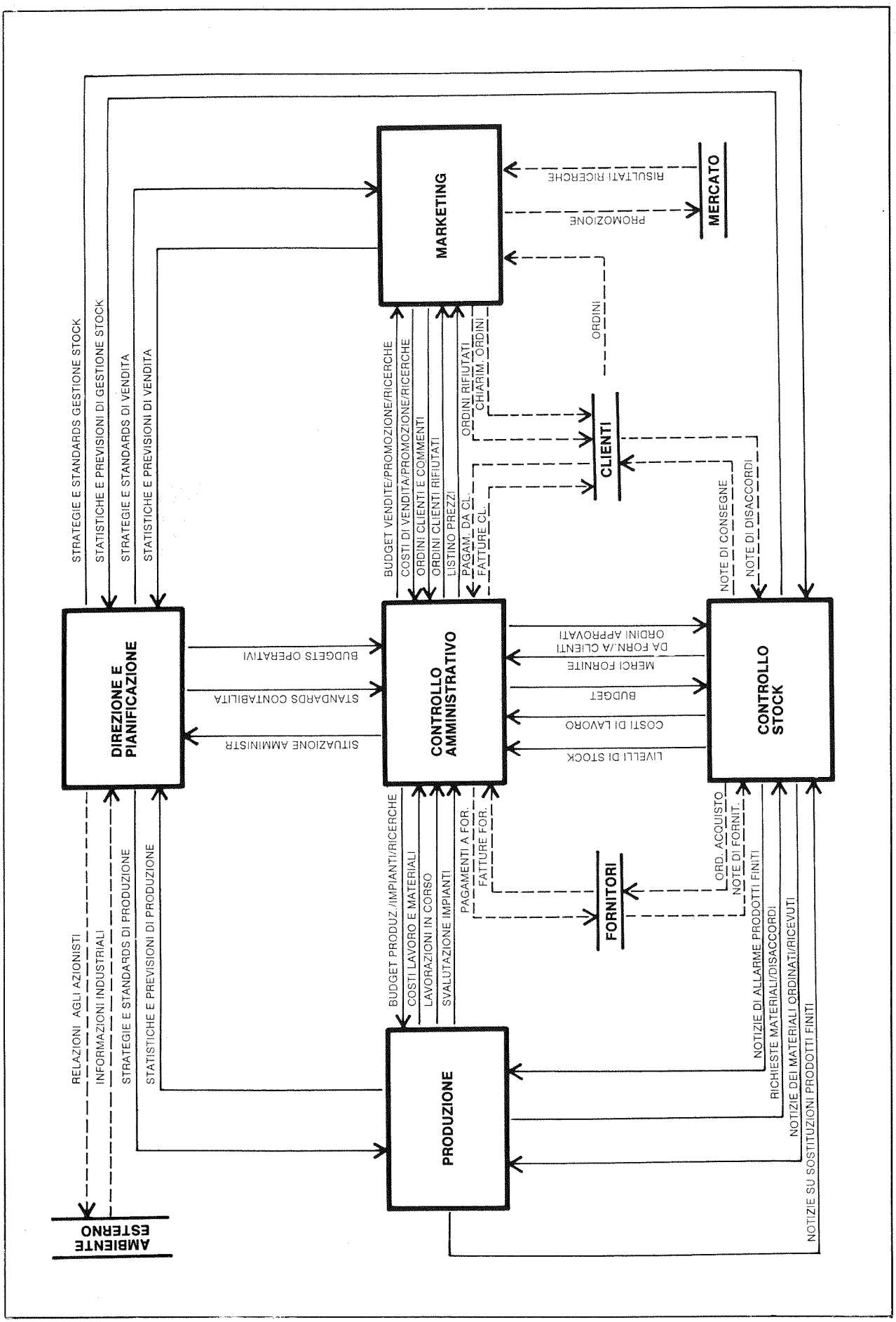


Fig. 2 - Esempio di interfaccia totale delle informazioni

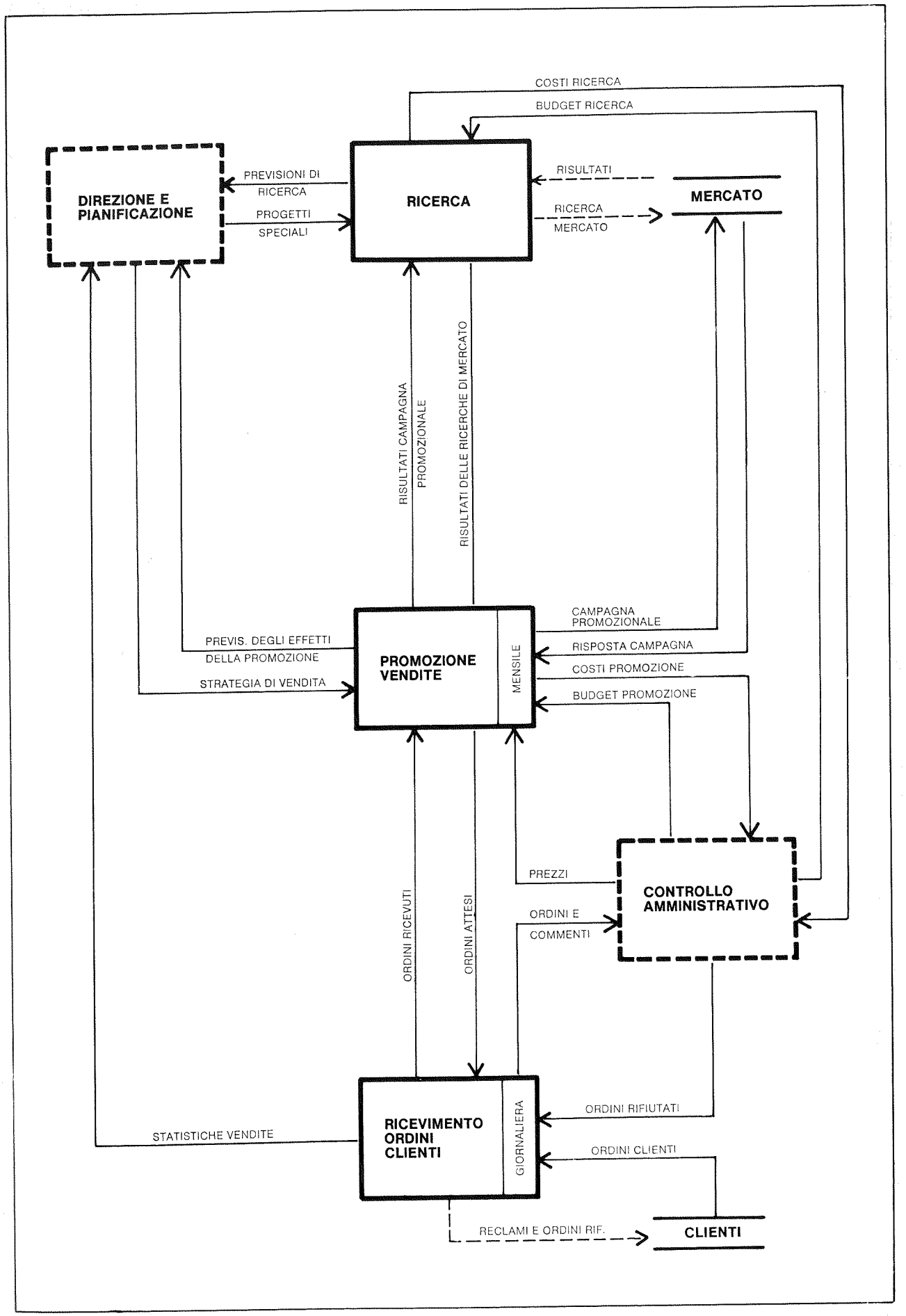


Fig. 3 - Esempio di inferfaccia funzionale (Marketing)

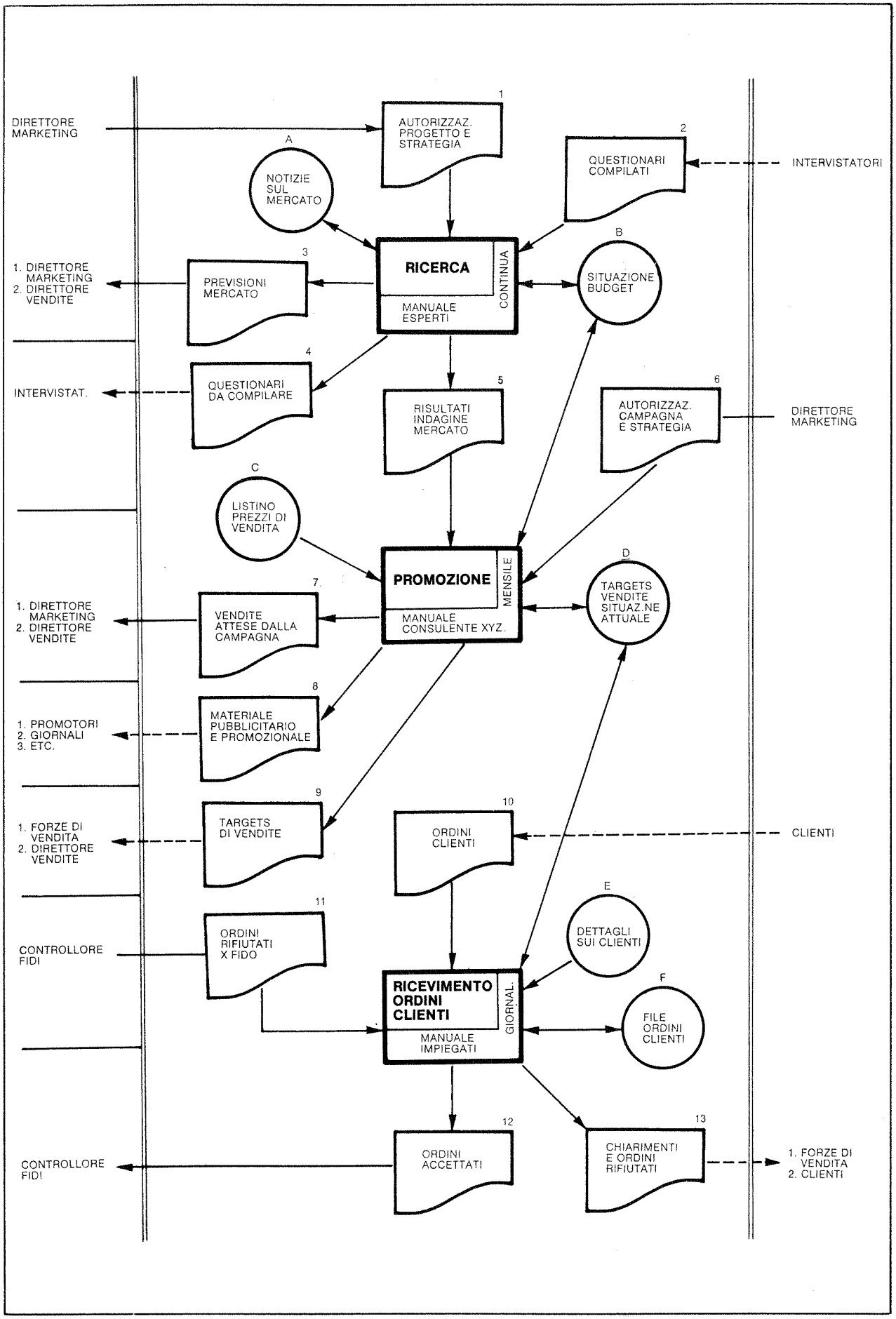


Fig. 4 - Esempio di prototipo funzionale (Marketing)

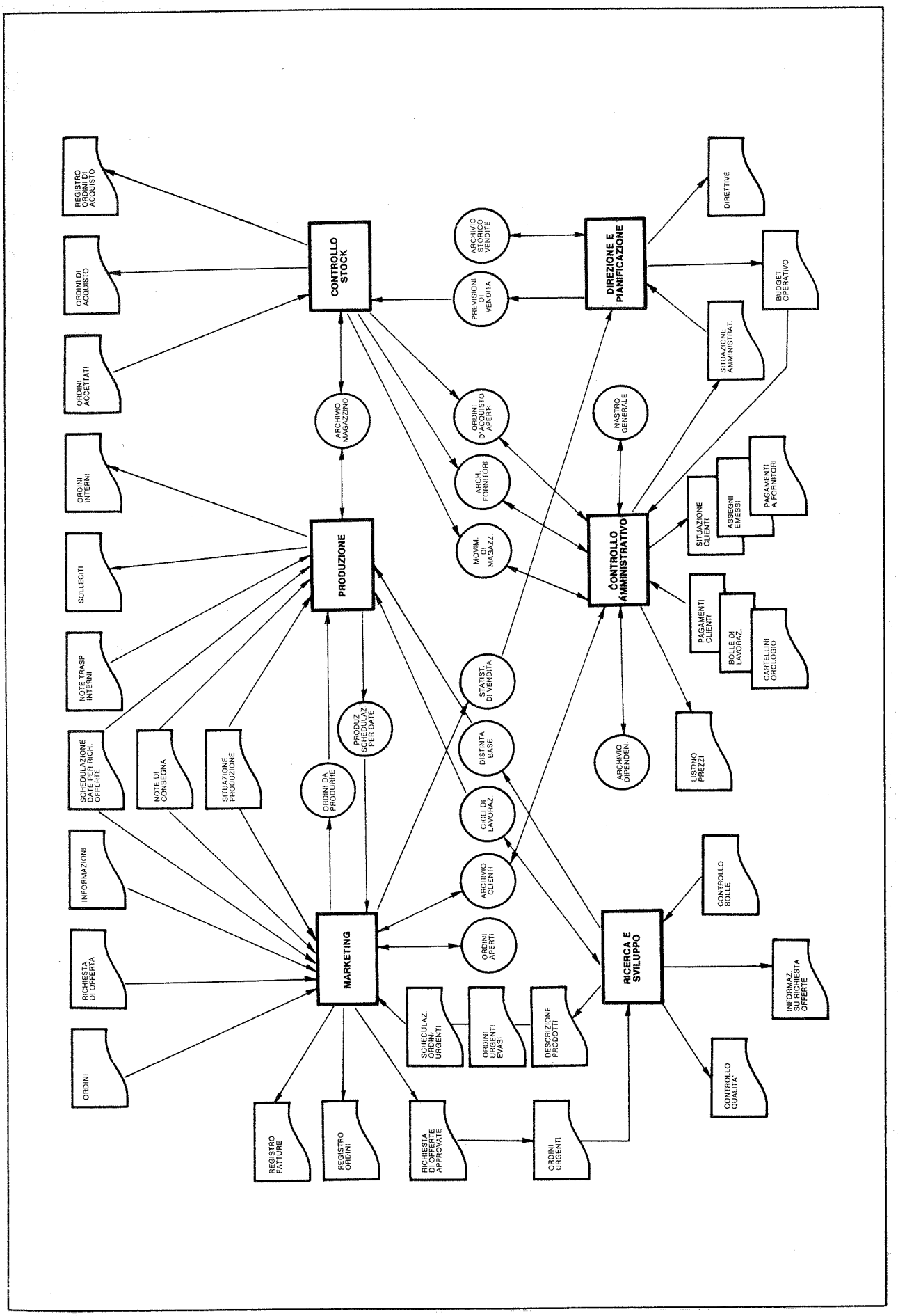


Fig. 5 - Esempio di Prototipo Totale

Inoltre sui Prototipi funzionali va evidenziato ai margini del diagramma « chi » nell'ambito della organizzazione aziendale fornisce i Veicoli dati e chi li utilizza.
La figura 4 mostra un esempio di Prototipo funzionale.
Completati i disegni dei Prototipi funzionali si passa alla operazione finale di questa fase che consiste in una implosione, cioè in una sintesi unificante di tutti i Prototipi funzionali che conduce al Prototipo totale che è il modello del sistema informativo totale dell'azienda.
La figura 5 mostra un esempio di Prototipo totale.

6. Specifiche del sistema

La documentazione del Sistema Informativo si realizza contemporaneamente allo svolgersi delle fasi del metodo fin quì illustrate, tuttavia l'analista deve dare un assetto formale a tutto il lavoro svolto allo scopo di informare l'alta direzione sui risultati dello studio, sul piano di realizzazione proposto e realative risorse necessarie.
In altri termini, senza sommergere la direzione di dettagli, l'analista deve metterla nella condizione di verificare la rispondenza dello studio agli obiettivi fissati e di decidere: egli deve soddisfare le aspettative della direzione che sostanzialmente vuole conoscere:

- che cosa si otterrà
- quando si otterrà
- quanto costa
- quali sono i benefici

Le specifiche del sistema sono costituite da tre parti:

1. Proposta alla Direzione
2. Guida alla scelta dell'hardware
3. Descrizione dettagliata del sistema

Non riteniamo opportuno illustrare questi aspetti in questa sede, non perché essi non assumono una particolare importanza, ma perché abbiamo voluto limitare questo lavoro alla metodologia di disegno di un sistema informativo, convinti che le difficoltà maggiori nel raggiungimento di questo obiettivo siano insite soprattuto nel lavoro di analisi.

7. Conclusione

Il metodo BISAD che abbiamo illustrato in queste pagine offre gli strumenti che consentono di progettare quella rete di comunicazioni ed informazioni che mette in grado la direzione aziendale di gestire e guidare l'azienda stessa verso i suoi obiettivi.
In paticolare i vantaggi che derivano dall'applicazione del metodo BISAD possono riassumersi nella più efficiente e tempestiva elaborazione delle informazioni destinate alla direzione.
Assolvere a questo imperativo significa poter controllare meglio l'azienda, migliorare le communicazioni dentro e fuori l'azienda, sfruttare meglio le possibilità offerte dagli elaboratori elettronici, in una parola, gestire l'azienda in modo più redditizio.
Ritornando a citare K. Grindley e J. Humble: « Gli obiettivi tecnici di per se non sono meritevoli di essere perseguiti..., l'unico valido obiettivo per gli elaboratori è di supportare i conseguimenti di miglioramenti gestionali che sarebbero irraggiungibili e antieconomici senza di loro »

8. Bigliografia

C.F. GIBSON, R.L. NOLAN, *Managing the four stage of EDP growth,* Harward Business Review 1974 January - February.

K. GRINDLEY, J. HUMBLE, *The effective computer,* Londra Mc Grow-Hill - 1973

HONEYWELL I.S. *Building an information system - The BISAD method* - 1972.

Esperienze sull'impiego degli elaboratori nell'istruzione

GIOVANNA GAZZANIGA
LILIANA IRONI
Laboratorio di Analisi Numerica del C.N.R.
Pavia

1. Introduzione

Ci si propone, in questo articolo, di illustrare i principi e le possibilità di impiego del calcolatore nell'istruzione attraverso la descrizione di una sperimentazione concreta.

Si fa riferimento, allo scopo, all'attività svolta da alcuni anni presso il Laboratorio di Analisi Numerica del C.N.R. in Pavia nel settore della « Computer Assisted Instruction » (C.A.I.), avente come obiettivo l'acquisizione di concrete esperienze sull'efficacia dell'impiego del calcolatore nell'insegnamento di alcune discipline a livello universitario e medio superiore.

In tale sede, per avere la possibilità di dare una completa valutazione sulla validità di tecniche C.A.I., si è ritenuto opportuno mettere a punto due corsi: uno per sperimentare l'uso del calcolatore come strumento autonomo d'insegnamento, l'altro per valutare le possibilità di impiego del calcolatore come strumento d'appoggio ad un corso di tipo tradizionale.

A tale scopo si sono realizzati un corso completo di teoria ed esercizi per l'insegnamento del linguaggio di programmazione FORTRAN ed un corso di esercizi di algebra lineare e di analisi numerica. La scelta di questi due argomenti è dovuta sia alla loro caratteristica di essere facilmente schematizzabili e quindi particolarmente adatti ad un insegnamento programmato, sia alle locali esigenze di carattere didattico.

Entrambi i corsi sono stati sperimentati da parte degli studenti di Matematica e Fisica dell'Università di Pavia ed il corso FORTRAN è stato sperimentato anche su un gruppo di studenti della Scuola Media Superiore.

Il periodo di sperimentazione consente di esprimere un giudizio sostanzialmente positivo sul valore di una attività nel settore dell'istruzione assistita da calcolatore e sulle buone possibilità di impiego di corsi C.A.I. a livello universitario. Questa esperienza ha messo, inoltre, in luce che la qualità di un corso C.A.I. dipende in stretta misura, oltre che dai contenuti, anche dai supporti software e hardware a disposizione ed, in particolare, dal linguaggio implementatore usato. Per i corsi suddetti è stato utilizzato il linguaggio COURSE WRITER III.

In seguito per rispondere a diverse esigenze, in particolare di flessibilità nel colloquio studente-elaboratore e di « portabilità » su sistemi diversi, è stato sviluppato presso il L.A.N. di Pavia un sistema di software per il C.A.I. (denominato LANCAI).

Questo sistema, implementato in un linguaggio indipendente dalla macchina, e basato su un interpretatore del linguaggio COURSE WRITER, ha consentito di raggiungere un buon livello dei prefissati obiettivi di portabilità e flessibilità di colloquio. Il sistema LANCAI attualmente, oltre a gestire corsi esistenti, viene utilizzato per sviluppare nuovo materiale didattico.

2. Corso di fortran

Questo corso, con il quale si sono volute sperimentare tecniche C.A.I. per l'insegnamento del linguaggio di programmazione FORTRAN, è composto da 18 lezioni e da un gruppo di 8 esercitazioni.

L'unità base di una lezione è il « paragrafo », che può essere di due tipi fondamentali:

1) paragrafo interattivo
2) paragrafo descrittivo

Il « paragrafo interattivo » consta di una « sezione informativa », in cui si fornisce allo studente una nozione base; di una « sezione esplicativa », in cui la nozione viene illustrata con esempi; e di una « sezione conversazionale » in cui viene verificato l'apprendimento di quanto esposto mediante domande a cui lo studente deve rispondere.

Il « paragrafo descrittivo » è utilizzato per spiegare allo studente i motivi per cui certe nozioni vengono fornite; non viene richiesto, quindi, un dialogo con lo studente.

Particolare risalto è stato dato al problema dell'analisi della risposta: è sembrato molto importante,

infatti, poter avere sotto controllo il livello di apprendimento delle nozioni esposte. L'analisi della risposta errata deve quindi essere in grado di individuare il tipo di errore commesso (ad esempio, errata interpretazione di quanto esposto nella « sezione informativa », ignoranza di nozioni precedenti, o semplicemente stanchezza) in modo da venire in aiuto allo studente con spiegazioni supplementari, oppure con il rinvio a paragrafi precedenti oppure con il consiglio di sospendere la seduta, a seconda del caso.
Si è voluto anche evitare che lo studente apprendesse nuove nozioni senza avere ben assimilato le precedenti e per questo si è fatto precedere ad ogni lezione, tranne la prima, un gruppo di esercizi di ripasso sulla lezione precedente, che lo studente deve risolvere in modo corretto per poter accedere alla nuova lezione.
Le « esercitazioni » sono sessioni al terminale alle quali l'allievo accede dopo aver seguito un primo gruppo di lezioni che gli forniscono le nozioni indispensabili per lo svolgimento di un esercizio significativo.
Ogni esercitazione è suddivisa in tre parti.
Nella parte centrale vengono svolte, le seguenti attività:

- Presentazione del testo di un problema.
- Breve descrizione del metodo di risoluzione predisposto dall'autore e stampa del flow-chart del programma che dovrà essere codificato.
- Stesura del programma. Questa avviene in forma « guidata », cioè lo studente viene seguito e corretto istruzione per istruzione.
- Preparazione dell'INPUT. L'elaboratore propone allo studente un set completo di dati di prova nella forma prevista dalle istruzioni READ e FORMAT del programma.
- Stampa dell'OUTPUT. L'elaboratore stampa un facsimile del tabulato di OUTPUT che si sarebbe ottenuto eseguendo il programma appena codificato con i dati di INPUT precedenti.

Nella terza parte di ogni esercitazione viene proposto il testo di un problema che lo studente dovrà cercare di risolvere da solo, in altra sede, senza la guida dell'elaboratore. Anche in questo caso allo studente vengono date alcune indicazioni sul procedimento risolutivo da scegliere. Scopo di queste indicazioni è di permettere allo studente di confrontare la propria soluzione con quella predisposta dall'autore, che gli verrà presentata nella prima parte dell'esercitazione successiva.
Il flow-chart di fig. 1 illustra lo schema generale di una lezione.

3. Corso di esercizi di algebra lineare e di analisi numerica

Questo corso vuole essere un esperimento di impiego di tecniche C.A.I. come ausilio all'insegnamento tradizionale.
Esso è costituito da numerosi esercizi sui seguenti argomenti di algebra lineare e di analisi numerica: matrici; determinanti; sistemi di equazioni lineari; autovalori e norme di matrici e vettori; risoluzione di sistemi di equazioni lineari con metodi diretti; risoluzione di sistemi di equazioni lineari con metodi iterativi; argomenti introduttivi all'analisi numerica (sistemi di numerazione, aritmetica decimale in virgola mobile, errori e loro propagazione).
Requisito indispensabile per accedervi è la conoscenza delle definizioni e degli enunciati dei teoremi riguardanti gli argomenti trattati negli esercizi. È anche possibile, comunque, un'utilizzazione più generale del corso, come corso quasi autonomo, in quanto le nozioni fondamentali possono venire ricordate durante lo svolgimento delle sessioni al terminale. Lo studente può, infatti, disporre di un glossario e richiedere le definizioni e le regole di cui necessiti. Per ogni argomento si è inoltre previsto che lo studente possa richiedere anche una lista di fonti bibliografiche relative a quanto appena trattato.
Il corso si propone di offrire allo studente la possibilità sia della verifica dell'apprendimento di alcune nozioni teoriche, sia di familiarizzarsi con procedimenti di calcolo. Si sono utilizzati, quindi, due modelli diversi di esercizi:
A. esercizi a carattere concettuale, che hanno come scopo fondamentale la verifica di nozioni teoriche apprese per altra via.
In questo tipo di esercizi viene, quindi, quasi completamente trascurato l'aspetto più propriamente numerico del problema in esame, per evitare che, impegnato in conti laboriosi, lo studente perda di vista la continuità logica dell'esercizio. A questo scopo è stata predisposta una serie di « servizi », quali il calcolo di determinanti o di minori di matrici, che lo studente può richiedere al terminale in ogni momento.
B. esercizi di carattere tecnico, che si propongono di rendere più familiari allo studente alcuni procedimenti algoritmici.
Quest'ultimo tipo di esercizi ha in generale la seguente struttura:
1. Si verifica l'applicabilità del metodo numerico in esame.
2. Si richiede di risolvere, con un procedimento « guidato » un primo problema, in cui i passi dell'algoritmo sono estremamente dettagliati.
3. Si propone allo studente un problema di tipo analogo al precedente che deve essere risolto in modo « autonomo ». In caso di risposta errata o incerta la diagnosi e la correzione dell'errore vengono effettuate con un procedimento di tipo guidato.
4. Nel caso in cui l'algoritmo è più complesso si è ritenuto opportuno fornire, al termine di ogni gruppo di esercizi relativi ad uno stesso metodo, una schematizzazione del procedimento seguito mediante un approssimativo diagramma di flusso.

Per dare maggiore libertà allo studente e tenendo conto che il corso ha come scopo principale la verifica del livello di conoscenza di nozioni teoriche apprese in modo tradizionale che possono essere più o meno estese ed influire, quindi, in modo notevole sul rendimento del soggetto, si è pensato di non strutturare il corso in lezioni, ma di suddividerlo in base ai diversi argomenti trattati: all'allievo è lasciata

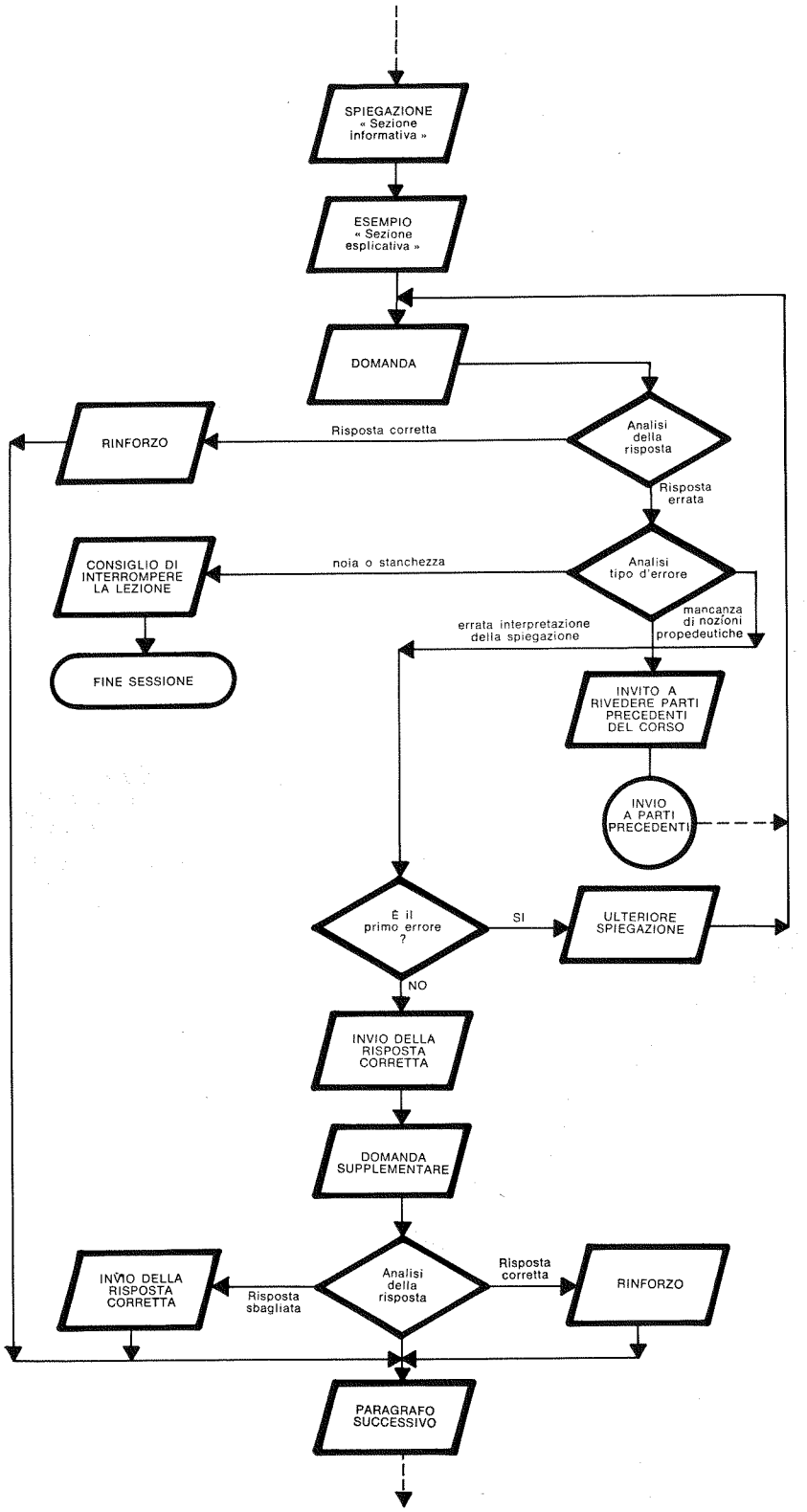


Fig. 1 - Schema generale di una lezione del corso di FORTRAN

la possibilità di interrompere la seduta al terminale quando lo ritenga opportuno. La seduta successiva riprende dal punto in cui il corso è stato sospeso.
La valutazione del lavoro svolto viene effettuata con i seguenti criteri:
— se lo studente fornisce le risposte esatte, si sono previste domande di controllo atte a verificare che la risposta corretta non sia frutto di memoria o sia dovuta al caso. Se lo studente dimostra di conoscere effettivamente la materia, lo si approva esplicitamente e lo si invita a proseguire.
— Se lo studente non sa rispondere, si spezza la domanda in sottodomande. Queste sono presentate in cascata secondo uno schema lineare. In caso lo studente dimostri molta incertezza nello svolgimento dell'esercizio gli si propone un esercizio di rinforzo dello stesso tipo del precedente, ma con minori difficoltà. Se lo studente non è in grado di rispondere neppure all'esercizio di rinforzo si suppone che tale atteggiamento nasca da effettiva scarsa conoscenza dell'argomento o da scarso impegno per il lavoro che sta facendo e, di conseguenza, lo si invita ad interrompere la lezione. La fig. 2 illustra lo svolgimento di un esercizio « tipo ».
Le difficoltà incontrate nella stesura di questo corso sono dovute principalmente alla struttura del sistema COURSEWRITER che, efficace per la presentazione del materiale didattico, è risultato però carente quando è sorta l'esigenza di utilizzare il terminale anche come « deskcalculator ».

4. Sperimentazione didattica

Scopo fondamentale del progetto illustrato era lo studio del comportamento degli studenti di fronte ad un esperimento C.A.I. e l'esame dell'efficacia di questo metodo di insegnamento comparato con i metodi didattici tradizionali.
Il periodo di sperimentazione è stato diviso in due momenti. Un primo momento è stato dedicato alla messa a punto dei corsi, cioè all'individuazione di tutti gli elementi atti a rendere più efficiente il corso; in particolare a migliorare il testo delle lezioni sia nell'impostazione generale che nei dettagli e ad organizzare in maniera più organica il corso nel suo complesso. Per raggiungere questo scopo si è scelta una sperimentazione su scala ridotta, cioè su un numero limitato di studenti che sono stati seguiti mediante l'osservazione del loro comportamento durante le sedute ai terminali (per rilevare momenti di noia o di stanchezza ed eventuali disturbi fisici), e mediante l'analisi dei tabulati prodotti durante le lezioni: ogni errore è stato vagliato ed interpretato e sono state fatte delle statistiche per valutare il rendimento medio.
Un secondo momento, successivo al perfezionamento del corso sulla base delle indicazioni precedenti, è stato dedicato ad una sperimentazione su larga scala: i corsi sono stati resi accessibili a tutti gli studenti dei corsi di laurea in Matematica e in Fisica. Come strumento di indagine, in questo secondo momento, si è usato un questionario appositamente predisposto e liberamente compilato dagli studenti.
La reazione degli studenti di fronte a questo tipo di apprendimento è stata, nel complesso, positiva. L'entusiasmo iniziale, nato dalla novità del metodo didattico e dalla possibilità di dialogare con il computer, non è, in generale, diminuito durante l'esperimento. È stato molto apprezzato il modo individualizzato con cui viene impartito l'insegnamento e la possibilità di adeguare il ritmo di apprendimento alle proprie capacità e conoscenze. Utile è sembrata anche la possibilità di costruire con i tabulati delle lezioni un testo personale che lo studente stesso ha collaborato a scrivere e che gli è quindi molto più familiare di un testo tradizionale.
Si è poi verificato, mediante il confronto con altri studenti che avevano seguito analoghi corsi di tipo tradizionale, che le nozioni apprese con i corsi C.A.I. rimangono maggiormente impresse, perché hanno richiesto per la loro assimilazione un maggior lavoro di concentrazione da parte dell'allievo.

5. Caratteristiche di un linguaggio di programmazione orientato al C.A.I.

L'esperienza acquisita durante la realizzazione e la sperimentazione dei corsi programmati descritti in precedenza ha permesso di identificare alcune caratteristiche di cui un linguaggio di programmazione orientato al C.A.I. dovrebbe godere.
Esso dovrebbe essere, innanzitutto, facile da usare anche da chi è poco esperto di tecniche di programmazione ed il più possibile « autodocumentato ». Dovrebbe essere flessibile in modo da potersi adattare bene all'insegnamento di diverse discipline, dovrebbe permettere di richiamare routines scritte in linguaggi di programmazione differenti. Dovrebbe essere « intelligente », in modo da riconoscere la risposta dello studente anche quando questa non è scritta secondo un formulario prefissato. Dovrebbe godere di un buon grado di portabilità, in modo da assicurare l'impiego anche su calcolatori con caratteristiche o configurazioni diverse da quelle su cui è stato realizzato e dovrebbe occupare poco spazio in memoria, in modo da poterne prevedere l'uso anche su calcolatori con modeste dimensioni di memoria. Infine dovrebbe essere a struttura modulare, così da garantire l'estendibilità o l'aggiornamento con poche modifiche.
Il linguaggio COURSEWRITER gode di alcune di queste caratteristiche, quali, ad esempio, la semplicità e l'autodocumentazione; mentre è molto carente rispetto alle altre caratteristiche che sono state esposte. In particolare esso si dimostra inadatto a sfruttare la potenzialità di calcolo della macchina che gestisce il corso di istruzione, mentre questa possibilità sembra molto importante per l'insegnamento di alcune discipline, quali l'Analisi Numerica e l'Algebra Lineare.
Apparve allora naturale cercare di realizzare autonomamente un sistema di software per il C.A.I. che godesse di quelle caratteristiche che erano state individuate come particolarmente interessanti: si è quindi pensato ad un sistema di software implementato in un

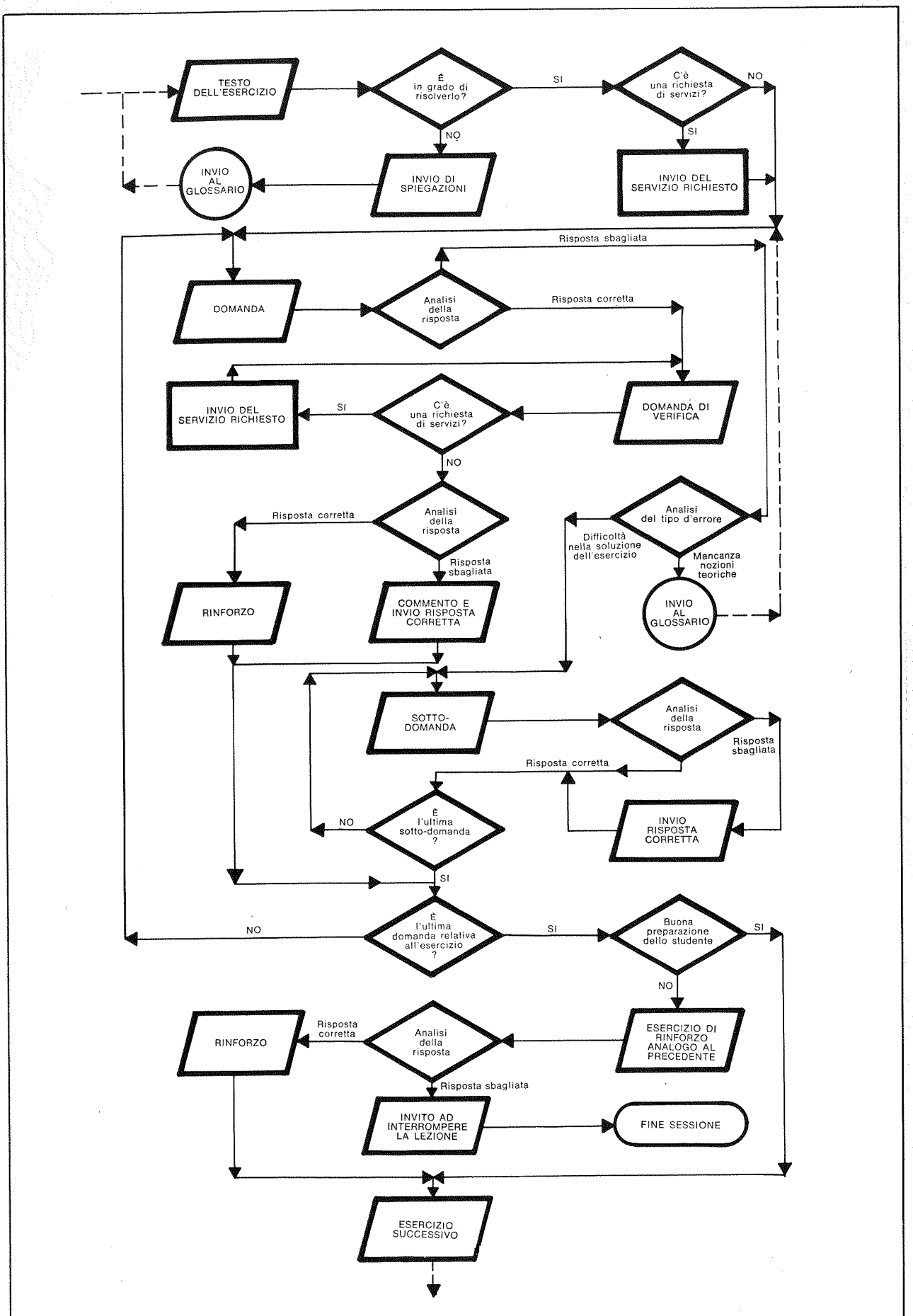


Fig. 2 - Schema di svolgimento di un esercizio del corso di esercizi di algebra lineare e di analisi numerica.

```
          eserc1
          Calcolare un valore approssimato della radice positiva dell equazione:
                    2
              (X/2) -SEN X=0
          Per proseguire premi il tasto RETURN
=

          Quale tra i seguenti metodi vuoi applicare?
            1- Newton
            2- Bisezione
            3- Secanti
          Rispondi: 1, 2, 3
= 1

          Scrivi la formula iterativa del metodo di Newton per una
          generica funzione f(x).
          x(n+1)=?
          (Scrivi f(x(n)) per indicare il valore di f(x) calcolato nel punto x(n))
=  x(n+1)=x(n)-f(x(n))/f'(x(n))

          Esatto.
          Ti interessa vedere una tabulazione della funzione?
=  si

          Scegli l'intervallo [a,b] di tabulazione ed il passo h.
          a=? ; b=? ; h=?
=  a=0.5;b=2;h=0.2

               X            F(X)
         0.500000     -0.41692554
         0.700000     -0.52171769
         0.900000     -0.58082691
          1.10000     -0.58870736
          1.30000     -0.54105818
          1.50000     -0.43499497
          1.70000     -0.26916479
          1.90000     -0.43800061E-01
          2.00000      0.90702576E-01
          Le informazioni che hai avuto ti hanno dato delle indicazioni che
          ti permettono di proseguire?
=  si

          Puoi passare a verificare le condizioni sufficienti di convergenza
          del metodo. Data l'equazione  f(x)=0  , con f(x) continua con le
          sue derivate prima e seconda in un intervallo [a,b]:
          1) Come deve essere il segno del prodotto f(a).f(b) ?
=  negativo

          In effetti nell'intervallo da te scelto accade proprio questo.
          2) Come deve essere la derivata prima nell'intervallo considerato?
=  diversa da zero

          Bene!
          Calcola e scrivi f'(x)  (Ricorda che f(x)=(x/2)**2-sen(x))
=  x/2-cosx

          Esatto
          Trova allora un intervallo [a',b'] contenuto in [a,b] in cui f'(x)
          e' diversa da zero e tale che f(a').f(b')<0 .
          a'=? ; b'=?
=  a'=1;b'=2

          Il tuo intervallo non soddisfa contemporaneamente alle tre condizioni:
          a- [a',b'] contenuto in [a,b]
          b- f'(x) diversa da zero in [a',b']
          c- f(a').f(b')<0
          Ti scrivo una tabulazione di f'(x) nello stesso intervallo [a,b] e con
          lo stesso passo h della tabulazione di f(x).
          X            F'(X)
       0.500000     -0.62758256
       0.700000     -0.41484218
       0.900000     -0.17160996
        1.10000      0.96403893E-01
        1.30000      0.38250119
        1.50000      0.67926282
        1.70000      0.97884453
        1.90000       1.2732896
        2.00000       1.4161468
          Trova allora un intervallo [a',b'] contenuto in [a,b] in cui f'(x)
          e' diversa da zero e tale che f(a').f(b')<0 .
          a'=? ; b'=?
=  a'=1.5;b'=2

          Va bene.
```

Fig. 3 - Esempio di svolgimento di parte di una lezione del corso di esercizi di algebra lineare e di analisi numerica.

linguaggio il più indipendente possibile dalla macchina e con quelle caratteristiche di flessibilità di colloquio e di possibilità di impiego di routines di calcolo abbastanza complesse, che il COURSEWRITER non aveva.
Per concretizzare il progetto, presso il L.A.N. è stato sviluppato sul calcolatore HONEYWELL 6030 un sistema, funzionante in ambiente time-sharing e realizzato in FORTRAN, basato su una soluzione interpretativa del linguaggio COURSEWRITER.
La scelta del linguaggio FORTRAN, quale linguaggio implementatore doveva garantire un prodotto con un buon grado di portabilità e di possibilità di diffusione. Nella sua versione conversazionale, inoltre, consente un colloquio rapido e semplifica la scrittura di routines matematiche.
La decisione di usare la soluzione time-sharing è apparsa molto vantaggiosa per le prestazioni che poteva offrire, anche rispetto al suo costo: rapidità di risposta, possibilità di eseguire routines scritte in linguaggi di programmazione diversi, possibilità di accedere alla biblioteca scientifica del calcolatore.

6. Il sistema LANCAI

Il sistema LANCAI realizzato sembra raggiungere abbastanza bene gli obiettivi esposti. Esso è strutturato in tre moduli:

— un interpretatore del linguaggio COURSEWRITER, che garantisce l'utilizzabilità di un qualunque programma scritto in tale linguaggio, senza alcuna sostanziale modifica;
— un sottosistema per il trattamento dei comandi d'autore;
— un sottosistema per il trattamento dei comandi per la supervisione dei corsi.

Questi moduli fanno riferimento a due archivi su disco:

— un archivio CORSO, che contiene il materiale didattico scritto in COURSEWRITER;
— un archivio STUDENTI, che contiene le informazioni relative agli studenti che stanno seguendo un certo corso. Infatti quando uno studente si iscrive, il suo nome ed il codice a lui assegnato vengono registrati in una tavola dell'archivio STUDENTI. In questa tavola viene anche memorizzato l'indirizzo di un'area dell'archivio a lui riservata e che conterrà tutta la sua « storia ». Nel corso di ogni lezione le aree riservate allo studente vengono modificate in modo da poter contenere informazioni sulle risposte date da ogni studente e condizionare la scelta del cammino da seguire. Questo permette di rendere più personale il colloquio con lo studente in quanto è così possibile la realizzazione di corsi a struttura altamente ramificata all'interno dei quali è lo studente stesso che determina con le sue scelte un particolare svolgimento.

L'interpretazione del linguaggio COURSEWRITER consiste di un corpo principale a cui si agganciano numerose routines e svolge le seguenti operazioni:

● Operazioni di inizio collegamento: riconosce il codice dello studente che ha chiesto di accedere al corso, individua l'indirizzo del punto da cui lo studente deve riprendere e porta in memoria principale le aree che gli competono.
● Interpretazione delle istruzioni: il sistema analizza il codice operativo di ogni istruzione ed eventualmente lo esegue, secondo la logica di « salto implicito » propria del linguaggio COURSEWRITER.
L'analisi delle istruzioni avviene in modo fondamentalmente sequenziale, lo studente può comunque richiedere in ogni momento di modificare lo svolgimento del corso con un'istruzione di salto ad un punto diverso. Molta cura si è avuta nella realizzazione della routine che analizza la risposta dello studente. Infatti il confronto della risposta dello studente con le risposte previste dall'autore del corso viene reso più elastico mediante l'uso di opportune « parole chiave ». Inoltre la novità presente nel sistema consiste nel poter riconoscere la risposta dello studente e lavorare su di essa, eseguendo ad esempio dei calcoli o qualunque altro tipo di operazione, prescindendo da un confronto con risposte previste dall'autore del corso. L'autore può tuttavia, valutando il risultato delle operazioni, stabilire la correttezza o meno della risposta dello studente e dare dei messaggi opportuni. In questo modo lo studente è libero di dare la sua risposta senza dover scegliere fra un numero limitato di alternative, mentre non viene persa la principale peculiarità di un corso di istruzione programmata, cioè il fatto di poter valutare ed eventualmente correggere le risposte date.
● Operazioni di fine collegamento: il corso viene interrotto quando si incontra l'etichetta di fine corso oppure su richiesta esplicita dello studente; in questo secondo caso si provvede a memorizzare l'indirizzo dell'ultima istruzione eseguita, che diventerà il punto di rientro per il prossimo collegamento, e ad aggiornare sull'archivio STUDENTI le informazioni relative allo studente interessato. In fig. 3 viene riportata una parte del colloquio relativo allo svolgimento di una lezione di un corso di esercizi di Analisi Numerica in cui sono state utilizzate le tecniche sopra descritte.

Il sottosistema per il trattamento dei comandi d'AUTORE svolge le seguenti operazioni, su esplicita richiesta dell'autore di un corso:

— carica nuovo materiale didattico, scritto in linguaggio COURSEWRITER, su un file CORSO già disponibile;
— cancella una o più istruzioni;
— stampa parti del corso;
— sostituisce una o più istruzioni con altre, anche in numero diverso;
— modifica l'ordine di alcune istruzioni.

Il sottosistema per il trattamento dei comandi per la supervisione dei corsi consiste di alcune routines che eseguono le operazioni necessarie alla generazione del file CORSO e del file STUDENTI ed alla gestione di un corso di istruzione programmata.
Le operazioni che possono essere richieste a questo sottosistema sono:

— inizializzazione di un file CORSO e del file STUDENTI relativo;
— iscrizione di studenti al corso;
— cancellazione di studenti;
— copia di parti di corsi da un file ad un altro;
— stampa di alcune parti del file STUDENTI;
— modifica dell'area personale di uno studente;
— riorganizzazione del file CORSO;
— stampa di parti riassuntive del file CORSO.

I sottosistemi che costituiscono il sistema LANCAI non hanno ovviamente la necessità di risiedere contemporaneamente in memoria e quindi l'intero sistema ha un'occupazione di memoria abbastanza modesta, in media 12 K parole sull'elaboratore HONEYWELL 6030. Il sistema si dovrebbe quindi poter utilizzare anche su elaboratori di dimensioni limitate.

Apparirà senz'altro evidente a coloro che conoscono il COURSEWRITER che il sistema LANCAI conserva le caratteristiche di semplicità e di efficienza del COURSEWRITER stesso. Rispetto ad esso offre in più la possibilità di scrivere facilmente dei sottoprogrammi. Questa è senz'altro la parte più nuova e più interessante del sistema in quanto estende le possibilità del linguaggio, aggiungendogli le caratteristiche più utili a seconda dell'argomento trattato nel corso. Ogni nuovo sottoprogramma deve essere redatto in FORTRAN e va aggiunto, dopo essere stato compilato, al sottosistema interpretatore.

Ovviamente il sistema può essere ulteriormente ampliato con l'aggiunta, ad esempio, anche di nuovi codici operativi, qualora sorgesse la necessità.

Anche il problema della portabilità sembra efficacemente risolto. Il sistema LANCAI è stato infatti trasferito senza grosse difficoltà sull'elaboratore IBM/370 del Centro Universitario di Calcolo dell'Università di Palermo.

7. Conclusioni

Sulla base della sperimentazione descritta, si può ritenere che esistono buone possibilità di impiego delle metodologie C.A.I. a livello universitario.

Il problema di un impiego generalizzato di tali metodologie presenta però i limiti ed i problemi che sono stati messi in luce in un precedente articolo pubblicato su questa rivista [7].

8. Bibliografia

[1] BARCHIETTO, BERTOLUZZA, BRAMBILLA, GAZZANIGA, ITALIANI, LUNGHI, SERAZZI, TELLAROLI, *"Un corso FORTRAN programmato mediante elaboratore elettronico"*, Pubbl. n. 21, L.A.N. del C.N.R., Pavia 1971.

[2] GAZZANIGA G., IRONI L., *"Esercitazioni di algebra lineare e di analisi numerica programmate mediante elaboratore elettronico"*, Pubbl. n. 85, L.A.N. del C.N.R., Pavia 1974.

[3] GAZZANIGA G., IRONI L., *"Some applications of computer aided instruction to numerical analysis and programming languages"*, Proceedings of 2nd World Conference on Computers in Education, Marseille 1975.

[4] GAZZANIGA G., IRONI L., ITALIANI M., *"Un sistema di software per l'istruzione assistita da calcolatore"*, Congresso A.I.C.A. Genova 1975.

[5] GAZZANIGA G., IRONI L., ITALIANI M., *"An interpreter which improves Coursewriter flexibility and portability"*, A.C.M. Sigcue Bulletin *"Computer Uses in Education"* 1976.

[6] GAZZANIGA G., IRONI L., *"Il sitema LANCAI per l'istruzione assistita da calcolatore"*, Pubbl. n. 135, L.A.N. del C.N.R., Pavia 1976.

[7] ITALIANI M., *"Prospettive didattiche dell'informatica"*, Quaderni di Informatica, III, n. 2, 1976.

L'elaborazione elettronica nel settore bancario in Italia

GIOVANNI BOSISIO
Servizio Studi Economici e Ricerche di Mercato
Honeywell Information Systems Italia

1. Introduzione

Il sistema bancario si è andato caratterizzando sempre di più negli ultimi anni come fornitore di servizi specializzati alla clientela e come soggetto strategico di primaria importanza nell'ambito del sistema economico, in relazione alla attività di erogazione del credito.

Questo ruolo, unito al crescente volume di mezzi finanziari amministrati e alla imposizione da parte delle autorità monetarie di costanti controlli sulla loro attività, ha posto alle banche, da un lato l'esigenza di automatizzare la notevole mole di lavoro contabile-amministrativo e di migliorare qualitativamente il livello del servizio, dall'altro la necessità, pur esplicitata in misura limitata fino ad oggi, di una razionale programmazione della loro attività.

L'approccio a tali problemi ha avuto ed ha tuttora profonde implicazioni in termini di EDP che portano a individuare in questo settore una delle aree di maggiore penetrazione e sviluppo della informatica.

Scopo del presente articolo (1) è una analisi del livello di meccanizzazione del settore, tendente a quantificare il fenomeno e la sua dinamica evolutiva, e ad individuare e qualificare le caratteristiche specifiche della domanda con particolare riferimento alle problematiche applicative.

L'universo di riferimento è costituito da tutte le aziende di credito ordinario esistenti in Italia al settembre 1976.

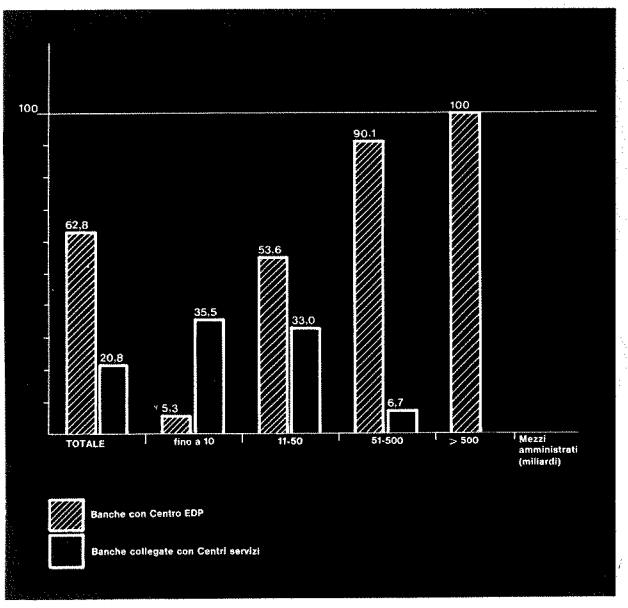
2. Livello e caratteristiche della meccanizzazione

Il settore bancario si pone, con riferimento alla realtà italiana, in una posizione di avanguardia per quanto riguarda la diffusione, l'intensità e la maturità della meccanizzazione EDP.

Delle 428 banche ordinarie considerate, 269 dispongono di un centro elettronico proprio, mentre 89 operano con procedure automatizzate, attraverso centri servizi.

Il modello di comportamento nei confronti del fenomeno in esame risulta condizionato principalmente dalle dimensioni aziendali (evidenziate in questa sede dal volume di mezzi amministrati). La meccanizzazione diretta è pressoché generalizzata per le banche con oltre 50 miliardi di mezzi amministrati. (fig. 1). Nella fascia inferiore (da 10 a 50 miliardi) la mecca-

Fig. 1 - Livello di meccanizzazione per classi di dimensione (dati percentuali)



(1) Il presente articolo fa riferimento allo studio « La meccanizzazione EDP nel settore bancario del Servizio Studi e Organizzazione della Honeywell Information Systems Italia, al quale si rimanda per gli aspetti metodologici. Si ricorda, in questa sede, che la data di riferimento delle informazioni è Settembre 1976 e che le stesse sono state reperite mediante due interviste - una intervista base rivolta a tutte le banche di credito ordinario e una intervista sui problemi applicativi sottoposta ad un campione rappresentativo di banche meccanizzate.

Regioni	Banche (1)	Centro Servizi (2)	2/1 %	Utenze EDP Dirette (3)	3/1 %
Piemonte, Liguria	36	10	27.7	23	63.9
Lombardia	78	17	21.8	55	70.5
Tre Venezie	53	17	32.1	28	52.8
Emilia Romagna	43	8	18.6	33	76.7
Toscana/Umbria	33	4	12.1	25	75.7
Lazio	28	8	28.6	16	57.1
Marche, Abruzzi	28	4	14.3	23	82.1
Puglia	36	1	2.8	24	66.7
Campania, Calabria Basilicata	49	9	18.4	20	40.8
Sicilia, Sardegna	44	11	25.0	22	50.0
Totale	428	89	20.8	269	62.8

Fig. 2 Banche-livello di meccanizzazione per regione

nizzazione EDP si qualifica ancora come esigenza largamente avvertita, soddisfatta però in misura rilevante attraverso il ricorso a centri servizi. Questo tipo di soluzione rappresenta spesso, in questa fascia, un primo passo verso la meccanizzazione diretta.

Per le banche più piccole le esigenze di meccanizzazione sono soddisfatte con l'utilizzo di strumenti più semplici (macchine elettrocontabili) o con ricorso a centri servizi, mentre la dotazione di un centro EDP proprio è limitata a casi sporadici.

Il livello di meccanizzazione risulta notevolmente differenziato in funzione delle aree territoriali, in relazione, principalmente, alla diversa struttura per dimensioni delle banche nelle varie aree; il livello di meccanizzazione relativamente basso nell'area meridionale dipende infatti dalla prevalenza in quest'area di banche piccole e medio-piccole.

Emergono però differenze regionali che vanno al di là di questo dato esplicativo. (fig. 2). Risulta particolarmente elevato infatti il livello di meccanizzazione nelle regioni centrali (Emilia Romagna, Toscana, Umbria, Marche, Abruzzi).

La notevole intensità della meccanizzazione nel settore è posta in risalto dalle dimensioni del parco installato: 372 sistemi con un valore di oltre 8 milioni di punti [2] (poco meno di 10 milioni, considerando anche la dotazione di terminali).

Caratteristica essenziale del parco è la sua concentrazione nelle banche di grandi dimensioni; il 30% dei sistemi e oltre il 70% del punteggio risulta installato nelle banche con oltre 500 miliardi di mezzi amministrati.

L'intensità della meccanizzazione in funzione delle dimensioni aziendali è evidenziata dall'andamento dell'indice punti medi per miliardo di mezzi amministrati. (fig. 3). L'indice, riferito ai punti di sistema, risulta decrescente al crescere delle dimensioni, ma, al di là di valori anomali per le due fasce estreme, giustificati dall'esistenza di rigidità dell'offerta verso il basso e da economie di scala oltre una certa dimensione, si mantiene entro un intervallo relativamente ristretto (tra 100 e 83).

L'andamento decrescente dell'indice è peraltro compensato, se si fa riferimento al punteggio totale, dell'incidenza dei terminali, progressivamente crescente in relazione diretta con le dimensioni aziendali.

[2] Il punto rappresenta un dollaro di locazione mensile, secondo i listini internazionali delle case costruttrici e costituisce l'unità di misura elementare del valore dei sistemi.

Fig. 3 - Intensità della meccanizzazione EDP per dimensione

Mezzi Amministrati (miliardi)	Banche meccanizz. (1)	Punti sist. (2)	Punti Totali (3)	2/1	3/1	Punti sistem. per miliardo	Punti sist. per miliardo
< 25	30	85	87	2.8	2.9	194	198
26-100	102	554	600	5.4	5.9	100	108
100-250	56	785	880	14.0	15.7	89	100
250-500	35	987	1095	28.2	31.3	83	94
500-1000	16	966	1237	60.4	77.3	91	116
> 1000	30	4770	5666	159.0	188.9	40	47
Totale	269	8147	9565	30.3	35.5	52	61

Mezzi amministrati (miliardi)	Banche con terminali	%*	Numero terminali	N. medio terminali per banca	Punti terminali	Punti term. / Punti sistema %
< 25	1	3.3	11	11	2	2.1
25-100	12	11.7	168	14	46	8.2
100-500	45	49.4	931	21	203	11.4
> 500	39	84.8	5351	137	1167	20.4
Totale	97	36.0	6461	67	1418	17.4

* % banche meccanizzate con terminali

Fig. 4 Terminali per classe dimensionale delle banche (mezzi amministrati)

Esiste cioè una consistente relazione lineare tra la dotazione hardware e il volume di mezzi amministrati, convalidata in termini analitici dalla retta di regressione [3]:
con

$$Y = -9.13 + 1.10X \qquad R^2 = 0.70$$
$$(0.05)$$

Y = Punti totali (x 100)

X = Volume mezzi amministrati (miliardi).

Il valore non elevato del coefficiente di determinazione (R^2) evidenzia tuttavia accanto alla dipendenza in media, una elevata variabilità del punteggio per miliardo tra le singole banche.
L'aggiunta di altre variabili indipendenti nella regressione (numero di sportelli, anno di meccanizzazione) non dà alcun significativo contributo aggiuntivo alla spiegazione della variabilità.
L'elevata variabilità sottintende evidentemente sia l'esistenza di diversità nel modello di meccanizzazione imputabile a fattori qualitativi, sia il mancato raggiungimento, per molte banche, di una relativa stabilità in rapporto al dimensionamento dei centri EDP. Quest'ultimo fattore risulta probabilmente assai rilevante nella fascia delle piccole e medio-piccole banche (con mezzi amministrati da 25 a 250 miliardi) per le quali la variabilità del rapporto punti per miliardo è particolarmente elevata.
In questa fascia molte banche sono tuttora in una fase di insufficiente maturità e di sviluppo accelerato dei centri e quindi lontane da una situazione di relativa stabilizzazione.
Un'ulteriore caratteristica della meccanizzazione nel settore è l'elevata anzianità (per oltre il 60% delle banche attualmente dotate di un centro EDP, l'inizio della meccanizzazione è antecedente al 1970) che determina un notevole grado di maturità del fenomeno. Ne deriva infatti, l'estesa acquisizione di un primo stadio relativamente stabilizzato di meccanizzazione e il soddisfacimento di esigenze di base di tipo contabile-amministrativo e quindi una predisposizione, almeno potenziale, ad orientarsi verso una domanda di informatica più evoluta.
Tale considerazione è applicabile alle banche con oltre 100 miliardi per le quali la meccanizzazione era un fatto generalizzato già nel 1970. Per le banche di dimensioni inferiori la meccanizzazione è invece, in larga misura di data recente (oltre i 2/3 dopo il 1970 e il 30% negli ultimi 4 anni).
Accanto alla elevata anzianità, va sottolineato il sensibile ritmo di aggiornamento dei centri EDP. Il 36% delle banche meccanizzate ha proceduto a modifiche dei centri o a nuove installazioni di sistemi nel periodo 1975-76.
Tale ritmo di aggiornamento si è mantenuto costante negli ultimi 3-4 anni qualificandosi come tendenza di medio periodo la cui intensità non sembra in fase di attenuazione.
In termini di evoluzione quantitativa, il parco sistemi si è sviluppato nel periodo 1971-76 ad un tasso annuo medio del 19.1% , con una accelerazione (21.2%) negli ultimi due anni e con valori superiori per la fascia delle grandi banche.
La struttura del parco installato risulta orientata verso sistemi di medie e grandi dimensioni (il 50% dei sistemi è superiore ai 10.000 punti e il 30% è superiore ai 25.000).
Sono identificabili fasce di dimensioni aziendali che definiscono uno specifico tipo di domanda. Fino a 100 miliardi e oltre 500 miliardi sono presenti rispettivamente i sistemi piccoli e i sistemi grandi in modo pressochè esclusivo, mentre nella fascia intermedia risultano presenti sistemi di diverse dimensioni.
L'analisi delle caratteristiche tecniche del parco conferma la tendenza verso sistemi medio-grandi e in particolare mette in luce la rilevante dotazione di dischi, in relazione allo sviluppo di applicazioni che esigono grandi quantità di Mbytes in linea.
I terminali che costituiscono l'elemento caratterizzante della attuale fase di concreto sviluppo dell'informatica nelle banche (il tempo reale di sportello) rappresentano una componente significativa dell'hardware installato (6500 installati per oltre un milione e mezzo di punti) (fig. 4).
Si tratta principalmente di normali terminali interat-

[3] Retta riferita alle 207 banche con volume di mezzi amministrati tra 25 e 1000 miliardi.

Mezzi Amministrati (Miliardi)	. Banche meccan. (1)	Banche con Term. (*)	Banche con Tempo Reale				Banche con T.R. iniziale
			Totale (**) (2)	$\frac{2}{1}$ %	20-50% sportelli	> 50% sportelli	
< 100	132	12	9	6.8	-	9	1
100-500	91	29	23	25.3	3	20	6
> 500	46	33	30	65.2	11	19	3
Totale	269	74	62	23.0	14	48	10

(*) Con rapporto Punti Terminali/Punti Sistema superiore al 10%

(**) Banche con Tempo Reale esteso ad almeno il 20% degli sportelli

Fig. 5 Tempo reale

tivi, mentre risulta limitata la presenza di terminali intelligenti e specializzati bancari, per i quali esistono peraltro significative prospettive di maggiore diffusione.

Circa un terzo delle banche esprime infatti un orientamento verso questo tipo di terminali che garantiscono una certa operatività anche in caso di interruzione del collegamento col centro; tale propensione risulta accentuata per le grandi banche (65%). 97 banche hanno attualmente una dotazione di terminali, per 74 delle quali il rapporto punti terminali/punti sistema è superiore al 10% [4].

L'impiego di terminali, rilevabile oltre la soglia dei 25 miliardi di mezzi amministrati, risulta notevolmente diffuso oltre i 100 miliardi e pressoché generalizzato oltre i 500 miliardi.

3. Tempo reale

Il tempo reale di sportello, che risponde alle esigenze di tempestività e di certezza delle informazioni, nelle operazioni che implicano un rapporto immediato con la clientela, costituisce oggi l'applicazione bancaria più sofisticata in fase di concreto sviluppo e con ulteriori prospettive di espansione.

Le banche dotate attualmente di tempo reale di sportello [5] sono 62, pari al 23% delle banche meccanizzate, delle quali 48 con una rete di terminali estesa ad oltre la metà degli sportelli.

La diffusione (fig. 5) risulta considerevole nelle grandi banche (65%), e ancora molto contenuta nelle banche di dimensioni inferiori nelle quali è però in atto un processo di espansione; in questa fascia infatti ben 7 banche hanno attualmente delle applicazioni in tempo reale in fase di realizzazione o ad un livello iniziale.

Oltre a porsi come importante fattore di razionalizzazione, il tempo reale si caratterizza attualmente come rilevante strumento di approccio alla clientela in termini di marketing bancario, stimolando quindi la competitività. La capacità dell'applicazione di incidere su questa duplice esigenza, oggi largamente avvertita da una fascia sempre più ampia di banche, spiega le prospettive di rapida espansione evidenziate in termini quantitativi dall'indagine: per i prossimi due anni ne è infatti prevista la realizzazione in altre 122 banche, con la generalizzazione nelle grandi banche e un'ampia diffusione sia nella fascia media (100-500 miliardi) che nelle banche di dimensioni inferiori. (fig. 6).

Il tempo reale tende quindi a qualificarsi come una esigenza generalizzata, estendendosi dalle grandi alle medie e piccole banche, con l'esclusione solo delle piccolissime (con meno di 25 miliardi).

Le applicazioni in tempo reale finora realizzate hanno riguardato quasi esclusivamente la gestione dei conti correnti e dei depositi per i quali l'immediatezza dell'informazione e del controllo sono fattori essenziali di razionalizzazione.

Nell'immediato futuro è prevista da parecchie banche l'estensione del tempo reale ad altre procedure. Va però considerato che per molte procedure bancarie, per le quali l'instantaneità della trasmissione non è un esigenza essenziale, potranno attuarsi modalità diverse di trasmissione dei dati.

La realizzazione del tempo reale implica, oltre ad una consistente dotazione di terminali, anche un notevole sviluppo dell'hardware di sistema.

L'indice punti per miliardo assume infatti valori sensibilmente più elevati per le banche con tempo reale, per le quali risulta inoltre sensibilmente più contenuta la variabilità dell'indice stesso (fig. 7).

Mezzi amministr. (Miliardi)	Banche meccanizz.	Campione		Stima	
		N.	Copert. %	%	N. Banche
100	132	35	26	43	57
100-500	91	25	27	60	54
> 500	46	46	100	24	11
Totale	269	106	40	45	122

Fig. 6 - Previsione di sviluppo del tempo reale entro il 1978

[4] Tale valore del rapporto è stato assunto come limite per distinguere una dotazione di terminali significativa da una dotazione iniziale o comunque non rilevante.

[5] Sono state considerate le banche con tempo reale esteso ad almeno il 20% degli sportelli.

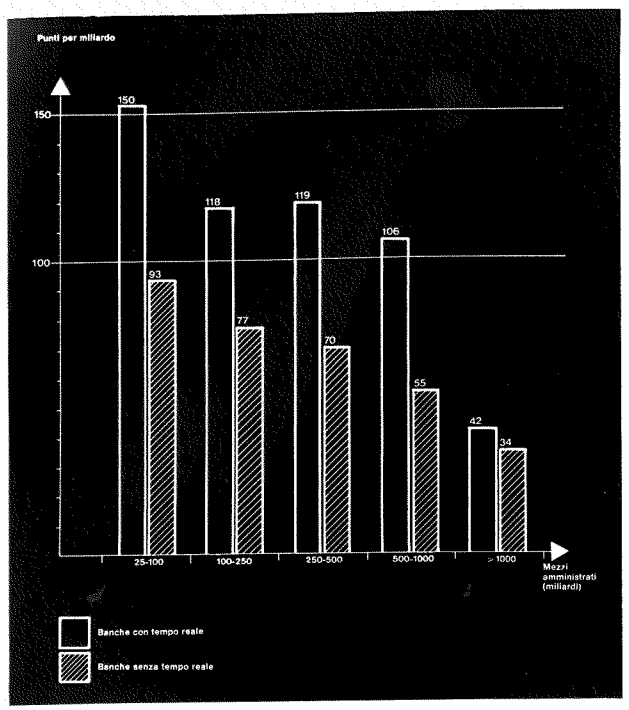


Fig. 7 - Confronto Banche con Tempo Reale e senza Tempo Reale (punti per miliardo di mezzi amministrati)

Questa tendenza applicativa costituisce quindi un fattore di relativa omogeneizzazione della dimensione dell'hardware in rapporto al volume di attività della banca.
Insieme con la prevista espansione del tempo reale, le ipotesi di decentramento dei centri EDP (la cosiddetta « informatica distribuita ») costituiscono il principale elemento condizionante il futuro sviluppo della meccanizzazione.
Si tratta di un discorso complesso per le vaste implicazioni che ne derivano sia in termini di ristrutturazione dei centri, sia in termini di mutamento della filosofia della banca.
Emergono a questo proposito orientamenti scarsamente innovativi. Il 92% delle banche intende semplicemente potenziare le strutture EDP esistenti; solo presso le maggiori banche sembra affermarsi (11%) l'orientamento verso l'informatica distribuita. (fig. 8). Si tratta però anche per queste banche, più che di un concreto orientamento, di una generica sensibilità al problema. In molti casi il giudizio negativo sul decentramento è motivato con gli elevati costi di riadattamento delle strutture informative. È un problema molto aperto e tuttora poco maturo per potersi tradurre in orientamenti concreti o in realizzazioni.

4. Aree applicative avanzate

In considerazione dell'elevato grado di maturità e delle esigenze di razionalizzazione della attività bancaria, uno dei nodi cruciali per lo sviluppo futuro dell'informatica nel settore è costituito dalla realizzazione di applicazioni indirizzate a fornire un supporto informativo e analitico alle funzioni decisionali e gestionali, che comportino quindi una più avanzata filosofia di utilizzo degli elaboratori.
Si tratta di applicazioni (analisi dei bilanci per la concessione di fidi, modelli di pianificazione finanziaria, di gestione della liquidità o di portafoglio, contabilità analitica dei costi, marketing bancario) che hanno già trovato all'estero uno specifico interesse e significative realizzazioni, mentre in Italia hanno incontrato fino ad oggi un interesse trascurabile.
Si riscontrano solo applicazioni sporadiche o studi sperimentali da parte di centri studi, non ancora trasformati in procedure operative sistematiche, ed anche in termini di orientamenti futuri solo un limitato numero di banche (attorno al 10%), prevalentemente di grandi dimensioni, esprime una propensione concreta in tal senso.
Più diffuso è l'interesse per le problematiche relative al trasferimento elettronico dei fondi. Gli orientamenti concreti riguardano però solamente l'adesione al progetto SWIFT per i regolamenti internazionali e l'estensione degli sportelli bancari automatici, intesa però principalmente come installazione di distributori automatici di cassa presso le aziende clienti. Altre iniziative avanzate di servizio diretto alla clientela, in fase di diffusione negli USA (impianti per la validazione della carte di credito presso i punti di vendita,

Fig. 8 - Orientamenti per lo sviluppo delle strutture EDP (entro il 1978)

Aree applicative	Stima percentuale banche					
	Totale		Interv. confid.	Mezzi amministr. (Miliardi)		
	%	σ	90%	< 100	100-500	> 500
Potenz. centri esistenti	92	2.0	93-98	97	97	89
Distrib. fisica e logica	5	1.6	2-8	-	4	11
Elab. « general purpose »	1	0.1	0.9-1.3	-	-	7
Minielaboratori dedicati	3	1.0	1-4	-	-	7
Terminali intell. o special.	38	4.0	31-45	23	44	67
Terminali a sola tastiera interattiva	17	4.0	11-23	20	16	11
Nessuna esigenza particolare	37	4.0	29-44	42	36	22

Aree applicative	Stima percentuale banche					
	Totale		Interv. confid.	Mezzi amministr. (miliardi)		
	%	σ	90%	< 100	100-500	> 500
Analisi bilanci per conc. fidi	13	3	8-18	9	16	20
Gestione fidi	54	4	47-61	17	80	67
Applic. di contabilità indust.	12	2	8-16	6	12	29
Applic. di ricerca operativa	7	2	3-10	6	4	15
Marketing bancario	11	3	6-15	11	8	13
Trasfer. elettron. di fondi	45	4	39-51	12	63	96
Partecip. consorzi EDP	19	3	14-24	5	32	33
Utilizzo packages applic. e/o prodotti programma	37	4	31-44	25	51	44
Utilizzo servizi e collab. esterne	9	2	5-12	3	12	18
Fonte preferenziale di soddisfacimento delle esigenze: Fornitore hardware	20	3	14-25	14	23	29
Case spec. di sofware	6	2	3-9	6	4	11

Fig. 9 - Orientamenti relativi alle applicazioni avanzate (entro il 1978)

sportelli automatici che comportino una estensione spaziale del servizio presso i poli di utenza) riscuotono un interesse, peraltro molto limitato, solo da parte delle grandi banche.

In conclusione, la realizzazione di sistemi informativi di tipo gestionale e di modelli e procedure per la programmazione delle attività, che dovrebbero fare delle banche organismi più efficienti, stenta a decollare, non solo in termini di implicazioni EDP ma anche come elaborazione di modelli analitici e operativi retrostanti. Questa situazione di stasi va fatta risalire principalmente alla struttura stessa del sistema bancario italiano. Si tratta innanzitutto di un mercato parzialmente protetto in cui i margini di competitività, con le conseguenti spinte alla razionalizzazione, sono piuttosto ristretti. In questi anni, caratterizzati dalla crisi dell'intero sistema economico, le banche hanno inoltre rappresentato un'area di privilegio, non vedendo messi in discussione in modo sostanziale i margini di redditività.

Questa realtà ha determinato l'assenza di grossi stimoli in direzione di una razionalizzazione del modello decisionale-gestionale - con riferimento, ad esempio, ai criteri di erogazione del credito o al controllo analitico dei costi e dei ricavi - permettendo la sussistenza di un modello scarsamente ancorato a supporti informativi ed analitici razionali.

In termini di sviluppo dell'informatica nel settore ne è derivato, sia con riferimento al passato che come prospettiva a breve termine, l'orientamento a proseguire il processo di razionalizzazione delle attività contabili-amministrative, cui in ultima analisi può ricondursi anche il tempo reale di sportello, piuttosto che la propensione verso interventi con un impatto diretto di tipo gestionale.

Certamente però le esigenze di razionalizzazione dell'apparato produttivo italiano poste dalla crisi e l'importanza fondamentale, in questo contesto, di una efficiente azione da parte del sistema bancario, potranno mutare nei prossimi anni i termini del discorso, rendendo necessaria per le banche una maggiore razionalità nei modi di operare, ai fini non solo di una qualificazione del loro ruolo ma anche del mantenimento della redditività aziendale stessa.

5. Conclusioni e prospettive

Le considerazioni fin qui svolte, portano alle definizione di fasce demensionali che qualificano diversi modelli di domanda EDP attuale e futura.

Per le grandi banche (oltre i 500 miliardi), caratterizzate da un elevato grado di maturità della meccanizzazione, la generalizzazione e l'ampliamento del tempo reale, già largamente diffuso, rappresenta la principale fonte di crescita del parco a breve termine. A medio termine il fattore decisivo per l'ulteriore sviluppo è costituito dalla capacità dell'utente di sviluppare applicazioni avanzate.

Le banche medie e piccole restano caratterizzate dall'esigenza di adeguamento dell'hardware a una dimensione di equilibrio e dall'orientamento verso la realizzazione del tempo reale. Sembra in atto in questa fascia un processo di imitazione del modello di domanda EDP proprio delle grandi banche.

Esiste infine una fascia di piccolissime banche, orientativamente al disotto dei 25 miliardi di mezzi amministrati, per le quali le esigenze di meccanizzazione continuano a porsi in termini semplificati. La definizione di un limite preciso è peraltro impossibile senza riferimento a considerazioni specifiche sulla struttura, l'ambiente, le modalità gestionali e operative.

In termine di prospettiva è possibile estrapolare dalle tendenze evolutive passate e tutt'ora in atto delle previsioni a breve-medio termine.

Il ritmo di crescita del parco si è mantenuto nell'ul-

timo quinquennio su livelli relativamente stabili, con una tendenza all'accentuazione nel periodo più recente.
Si è registrato un processo di adeguamento continuo dell'hardware agli accresciuti volumi di attività, che proseguità in modo inalterato per la fascia di banche medie e medio-piccole, mentre potrà risultare rallentato per le grandi banche, per le quali il semplice adeguamento ai volumi di attività si va avvicinando al punto di saturazione. L'espansione del tempo reale comporterà una notevole crescita dell'hardware nelle banche interessate (valutabile in oltre un milione di punti nei prossimi due anni, sulla base del divario medio oggi esistente tra banche con e senza tempo reale), oltre ad una consistente espansione del parco terminali, la cui entità dipenderà dal tipo di terminali prescelti, ma supererà probabilmente il milione di punti nel prossimo biennio.

In sintesi, si può prevedere a breve termine il proseguimento del ritmo di crescita di medio periodo (20%) in termini di hardware di sistema, con una accelerazione se si fa riferimento al parco totale.
A più lungo termine, lo sviluppo dell'informatica nel settore è condizionato da tutte le problematiche qualitative fin qui indicate.
Le esperienze straniere e le stesse esigenze latenti rilevate dimostrano l'esistenza di ampie aree di potenziale sviluppo, la cui praticabilità dipenderà dagli indirizzi organizzativi-gestionali delle banche, prima che dagli orientamenti specifici in termini EDP, se si considera che utilizzare gli elaboratori in un'ottica gestionale può implicare rilevanti mutamenti nella stessa filosofia gestionale e organizzativa. Molto dipenderà a questo proposito anche dalla capacità di stimolo da parte delle case fornitrici.

Resoconti e recensioni

Hanno collaborato a questa rubrica:
P. MAGGIOLINI, G. OCCHINI, G. RAPELLI, E. ROGNONI

Una nuova filosofia e una nuova tecnologia per l'elaborazione dei dati

Una nuova filosofia per l'impiego degli elaboratori elettronici e una nuova tecnologia per la realizzazione degli stessi sono state annunciate dalla Honeywell all'inizio dell'anno in corso, contemporaneamente alla presentazione di un insieme di nuovi prodotti hardware e software interessanti la serie 60 di elaboratori elettronici.

La nuova filosofia è quella dei sistemi distribuiti (Distributed Systems Environment o DSE) di cui si è già occupato con un suo articolo su questa rivista l'ing. Mike Canepa.

Diamo qui di seguito alcuni estratti dalla presentazione fattane, in occasione degli annunci di cui sopra, dall'amministratore delegato e direttore generale della Honeywell Information Systems Italia, ing. Carlo Peretti.

« Cominciamo col dire cosa significa elaborazione o informatica distribuita. In senso largo, qualunque organizzazione che elabori i suoi dati attraverso una molteplicità di unità elaborative si può dire che presenti una forma di elaborazione distribuita. Ma, se vogliamo darne una definizione più rigorosa, di elaborazione distribuita si può parlare per un'organizzazione in cui i dati vengano elaborati da due o più entità operanti *su base cooperativa*, e non su base semplicemente gerarchica come si ha nei casi in cui, anche se esistono terminali e calcolatori satellite, tutto il funzionamento del sistema dipende e viene gestito da un centro. In questo caso, anche se fisicamente si ha una distribuzione di risorse elaborative, il sistema è centralizzato.

È da notare che quando parliamo di entità elaborative non intendiamo semplicemente elaboratori nel senso classico, ma qualunque apparecchiatura che svolga una funzione attiva di elaborazione (modificando i dati, gestendo una base di dati) e non si comporti in maniera puramente passiva, come ad esempio un multiplatore di trasmissione dati.

In qualche misura l'elaborazione distribuita esiste già oggi, almeno in una forma embrionale. Alcuni utilizzatori hanno già collegato degli elaboratori in modo tale per cui una parte dell'elaborazione avviene localmente e un'altra parte a distanza, o per cui una parte delle transazioni viene trattata localmente mentre periodicamente i dati riassuntivi vengono inviati al centro perché esso provveda all'aggiornamento della base di dati centralizzata.

La tendenza che noi riteniamo stia delineandosi e nella direzione della quale abbiamo deciso di orientare la concezione dei nostri prodotti è quella di un aumento del grado di collaborazione fra entità elaborative e di una sua *invisibilità* (o, come si dice, *trasparenza*) *per l'utilizzatore*. Se i costruttori e gli utilizzatori saranno capaci di portare a compimento questa tendenza, sarà possibile in una qualunque elaborazione su uno qualunque degli elaboratori collegati fare riferimento a dati o utilizzare capacità elaborative indipendentemente dalla collocazione fisica di queste.

I vantaggi dell'elaborazione distribuita sono parecchi. Io vorrei evidenziare i seguenti:

— L'elaborazione avviene in prossimità del luogo in cui ha sede l'attività alla quale essa si riferisce. I tempi di risposta saranno perciò più brevi, e meno soggetti a ritardi.

— L'affidabilità del sistema aumenta dato che il malfunzionamento di un singolo componente non compromette il funzionamento dell'intero sistema.

— I costi di trasmissione, dal canto loro, possono essere ridotti in quanto la capacità di elaborazione viene portata a contatto dell'effettivo utilizzatore finale, riducendo il volume delle informazioni trasmesse.

— È possibile un miglior sfruttamento delle risorse e un'egualizzazione dei carichi di lavoro fra i vari elaboratori.

Questo naturalmente non significa che la filosofia dei

sistemi distribuiti verrà integralmente adottata in ogni organizzazione. Il secondo concetto che noi abbiamo cercato di seguire nel definire la nostra filosofia è infatti quello dell'opzionalità. Vale a dire che l'architettura dei sistemi deve essere tale da permettere all'utilizzatore di configurare i suoi sistemi come centralizzati o distribuiti a sua scelta e seguendo le esigenze della sua struttura organizzativa. Noi riteniamo che questa opzionalità e flessibilità di configurazione sia molto importante. A quanto sappiamo, vi sono utilizzatori che desiderano conservare un'impostazione centralizzata, eventualmente con calcolatori che svolgono funzioni ausiliarie rispetto all'elaboratore principale, altri che preferiscono un'organizzazione distribuita e altri ancora che desiderano un'impostazione centralizzata nella fase iniziale, in modo da ottenere uniformità nel lavoro di programmazione e nella definizione delle basi di dati, per passare più avanti a un'organizzazione di tipo distribuito.
L'importante, e desidero sottolinearlo, è che la scelta fra sistemi centralizzati e sistemi distribuiti sia dettata all'utente unicamente dalle sue esigenze organizzative e non gli venga imposta dalle caratteristiche dei sistemi.
Flessibilità dunque. Ma, ovviamente, la flessibilità non è tutto. Con una molteplicità di centri di elaborazione operanti ciascuno secondo proprie regole e procedure vi sarebbe un grosso rischio di confusione. Perciò noi ci siamo anche posti il problema di trovare una conciliazione fra libertà e flessibilità di configurazione e di accesso e la necessità di una standardizzazione nell'ambito del sistema e di un controllo e coordinamento delle attività di questo.
Pur senza voler anticipare quelli che saranno i nostri futuri programmi, credo di poter dire che attraverso l'uso di strutture unificate di dati, di regole standardizzate di trasmissione dati, di opportuni metodi d'accesso e di prestabiliti formati interni di messaggio sarà possibile realizzare sistemi che rispondano in maniera ottimale sia alle esigenze di flessibilità che a quelle di coordinamento e controllo.
Come ho già detto, l'idea dei sistemi distribuiti non è nuova e in qualche misura viene già realizzata oggi. Basi di dati locali vengono aggiornate in sito e poi trasmesse al centro. Inversamente, archivi vengono trasferiti dal centro a calcolatori periferici che eseguono elaborazioni su di essi. Dati vengono parzialmente elaborati da terminali intelligenti e poi trasmessi al centro. Quello che noi vediamo nel prossimo futuro è una semplificazione e una standardizzazione dei programmi e delle procedure speciali che sono oggi necessarie per svolgere questi compiti. Vediamo il generalizzarsi della possibilità che un elaboratore esegua la compilazione (cioè la traduzione da linguaggio simbolico a linguaggio macchina, che è un'operazione sempre impegnativa in termini di tempo e di memoria) di un programma e un altro elaboratore provveda all'esecuzione del programma stesso, che un elaboratore si rivolga a un altro per smistargli un certo lavoro, per fargli eseguire un programma o per ottenerne dati che sono colà presenti, e tutto questo senza creare problemi, trasparentemente come si diceva, e senza che addirittura il personale addetto a un elaboratore debba conoscere le caratteristiche e il modo di operare dell'elaboratore distante al quale si rivolge.
Naturalmente, e qui torno a parlare in termini di prodotti, tutto questo significherà l'ulteriore accrescersi e acuirsi di esigenze quali:
— l'esigenza di elaboratori aventi un buon rapporto costi/prestazioni e capaci sia di attività autonoma sia di collegamento ad altri elaboratori
— l'esigenza di più efficienti sistemi di gestione delle basi di dati
— l'esigenza di protocolli di trasmissione che assicurino standardizzazione e quindi corretti scambi di messaggi fra le varie apparecchiature
— l'esigenza di terminali più efficienti che permettano di collegare senza problemi le sedi periferiche della rete
— infine, più in generale, l'esigenza di tecnologie più avanzate che permettano la realizzazione di sempre nuove architetture di sistema.
Naturalmente non pensiamo di avere fin da ora la risposta a tutti i problemi. Quello che pensiamo di avere fin da ora è un approccio globale all'elaborazione distribuita che garantisca all'utilizzatore una maggiore flessibilità nel soddisfacimento delle sue esigenze. E quello che pensiamo di avere domani è una reale possibilità di realizzare sistemi distribuiti, attraverso basi di dati ad accesso comune, protocolli standardizzati, elaborazioni transazionali generalizzate e una piena compatibilità fra componenti del sistema.
Quello che non vogliamo è, d'altra parte, proporre cambiamenti improvvisi e radicali che avrebbero il duplice effetto negativo di mettere a repentaglio gli investimenti già fatti dall'utilizzatore e di costringere noi stessi e gli utilizzatori in schemi che potrebbero rivelarsi limitativi.
Perciò ci proponiamo di procedere con deliberata gradualità e, basandoci sulla esperienza nostra e degli utilizzatori, prepararci alla svolta che sicuramente i prossimi anni porteranno nel cammino dell'informatica. »
Per quanto riguarda la nuova tecnologia, si tratta di quella impiegata nel nuovo elaboratore elettronico Honeywell 66/85, il più potente sistema di elaborazione finora realizzato dalla Honeywell, caratterizzato da soluzioni, di avanguardia che consentono di ottenere eccezionali prestazioni di velocità.
Per illustrare queste soluzioni, occorre ricordare che la velocità di elaborazione di un segnale logico in una rete dipende sostanzialmente da due fattori e cioè:
1) dal ritardo intrinseco dei singoli elementi logici che costituiscono i nodi della rete, e
2) dalla lunghezza fisica dei collegamenti che connettono tali nodi, lungo il percorso compiuto dal segnale.
Per quanto riguarda il primo requisito, è stata impiegata la « logica a commutazione di corrente » (CML = Current-Mode-Logic), la più veloce oggi esistente,

con meno di 1 nanosecondo di ritardo per elemento. Nella soluzione specifica, tale velocità è stata ottenuta mantenendo un livello di dissipazione di energia eccezionalmente basso, ciò che ha permesso di integrare entro una singola tessera di semiconduttore (« chip ») un numero molto elevato di elementi logici. La tecnologia « micropackaging » costituisce invece la risposta al secondo requisito, cioè all'esigenza di ridurre al minimo le dimensioni fisiche del sistema di interconnessioni. A tale scopo, i « chip » sono montati direttamente, senza contenitore, su un supporto ceramico sul quale, in diversi strati sovrapposti, sono realizzati i percorsi conduttori che collegano i « chip » stessi. Il supporto ceramico ha la funzione, in questa soluzione, della convenzionale piastra a circuito stampato. Come risultato si ha una drastica riduzione dell'area occupata: una piastrina ceramica contiene centinaia di « chip », quanti ne possono stare su una piastra convenzionale 16 volte più grande.
La logica a commutazione di corrente ed il micropack-aging sono combinati in modo da sfruttare al massimo le potenzialità di ambedue, costituendo complessivamente la soluzione tecnologica più avanzata oggi disponibile.

G. R.

Giornata di studio su « L'informatica nelle piccole e medie aziende »

Milano, 25 gennaio 1977

Su iniziativa dell'AICA - Associazione Italiana per il Calcolo Automatico - e in collaborazione con la Honeywell Information Systems Italia, si è tenuta il 25 gennaio 1977 a Milano presso la FAST una giornata di studio su « L'informatica nelle piccole e medie aziende ».
Durante la giornata, alla quale hanno preso parte numerosi amministratori e dirigenti di piccole e medie aziende industriali e commerciali, sono state presentate cinque relazioni. Nella prima Mario Speranza, Direttore del Servizio Studi e Organizzazione della Honeywell I.S.I. ha esposto i risultati di un'indagine di mercato sulla diffusione in Italia degli elaboratori di piccole e medie dimensioni destinati ad applicazioni gestionali. Alla fine del 1975 si trattava di più di 4.500 sistemi, per il 47% impiegati nella piccola e media industria, per il 26% nel commercio e per il resto nelle amministrazioni locali, ospedali e banche. Il 67% di essi è localizzato nelle regioni dell'Italia settentrionale (il 33% nella sola Lombardia). Le soglie dimensionali per l'applicazione di elaboratori elettronici, sempre a quanto risulta da quest'indagine, stanno progressivamente scendendo, data anche la disponibilità sempre maggiore di modelli molto economici: nell'industria una notevole diffusione di piccoli elaboratori si ha ormai anche in aziende con meno di 100 addetti, e, nel commercio, in aziende con meno di 50 addetti.
I problemi della formazione tecnica e manageriale dei piccoli e medi utilizzatori edp sono stati affrontati, nella seconda relazione, da Fabrizio Martinoli, Direttore del Servizio Scuole Marketing della Honeywell I.S.I., con particolare riferimento alle nuove esigenze determinate dalla diffusione di forme di elaborazione interattiva.
L'esperienza diretta degli utilizzatori è stata portata nelle tre relazioni successive da Eugenio Quaglia, Presidente della Ceramica Adriatica S.p.A. di Potenza Picena (MC), il quale ha parlato dell'acquisizione ordini e fatturazione in tempo reale in un'azienda di produzione; da Giovanni Fossa, titolare della Mafotex di Cassano Magnago (VA), il quale ha parlato della gestione in tempo reale delle lavorazioni esterne e dei magazzini in un'azienda tessile e da Fabrizio Vecchi, Direttore Organizzazione della Fontana Luigi S.p.A. (viti, dadi e bulloni) di Veduggio (MI) il quale ha descritto il sistema informativo integrato e interattivo per il controllo gestionale realizzato presso la sua azienda. La giornata di studio è stata coordinata da Giulio Occhini della Honeywell I.S.I. ed è stata introdotta da Giampio Bracchi del Politecnico di Milano.

G. O.

Corso sui microelaboratori

Milano, 8-12 Novembre 1976

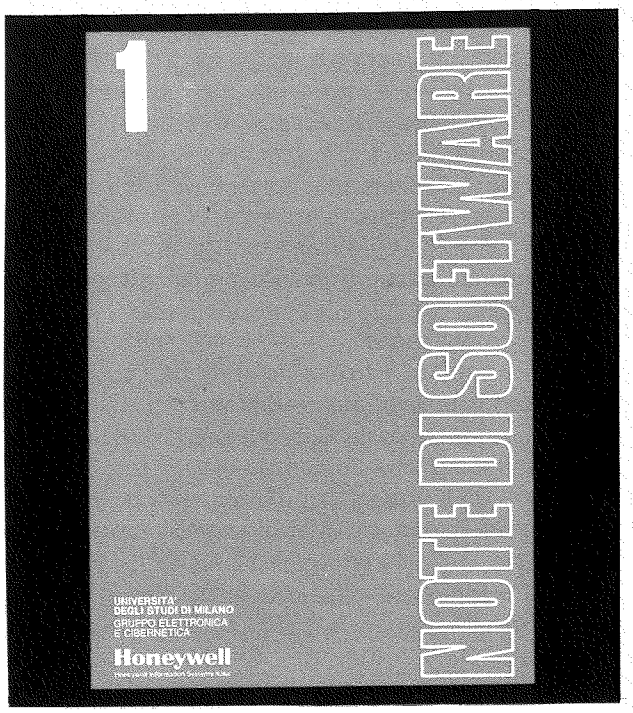
Organizzato dalla FOIST (Fondazione per lo sviluppo e la diffusione della istruzione e della cultura scientifica e tecnica) con l'adesione dell'ANIPLA (Associazione italiana per l'Automazione), si è tenuto nella settimana dell'8 novembre 1976 un corso su « Microelaboratori: progettazione hardware e software ».
Il corso, rivolto a persone già introdotte nell'argomento, aveva come obiettivo di fornire criteri concreti di progettazione hardware e software con microelaboratori, attraverso esempi di applicazione pratica di alcuni fra i più noti dispositivi esistenti sul mercato.
Nella prima giornata, il direttore del corso, Filippazzi, della Honeywell I.S.I., ha introdotto il tema illustrando le innovazioni che hanno portato alla nascita del microprocessore, con particolare riguardo alla evoluzione tecnologica. Sono state poi analizzate quantitativamente le tendenze tecnologiche e di costo per il prossimo futuro, evidenziando come i nuovi dispositivi abbiano influenzato le modalità di progetto e di fabbricazione degli apparati elettronici, con ripercussioni sul settore industriale e sulle attività professionali dei singoli, sempre più interdisciplinari ed orientate al lavoro di gruppo.
Nella stessa mattinata sono intervenuti Molinari, della Honeywell I.S.I., che ha parlato dell'architettura del microprocessore inteso come unità centrale di un

sistema di elaborazione e del sistema di controllo come organo supervisore della elaborazione, memorizzazione e trasferimento di dati e gestione del programma. Successivamente Brofferio, del Politecnico di Milano, ha trattato la struttura dei microelaboratori, i canali di collegamento tra unità centrale e memorie, le modalità di trasferimento dei dati, le periferiche, le interfacce, la tecnica di gestione delle interruzioni e degli accessi diretti alla memoria.
Nel pomeriggio, Minnaja, della Honeywell I.S.I., ha messo a fuoco i problemi che incontra il progettista, in particolare nella scelta tra dispositivi a larga integrazione, quali i microelaboratori, e quelli a bassa integrazione (logica sparsa), facendo una dettagliata analisi delle possibilità offerte dai microelaboratori ed allargando l'ottica di progetto ad applicazioni con più unità centrali.
Muzzani, dell'Olivetti, ha concluso le relazioni della prima giornata illustrando i sistemi di aiuto al progetto con microelaboratori, evidenziandone la potenza e soprattutto l'apporto fornito in sede di prova e di messa a punto dei programmi.
Nei quattro giorni restanti il corso è entrato nel vivo dell'argomento e sono stati sviluppati altrettanti progetti su applicazioni particolari analizzando sia gli aspetti hardware che software e vedendo concretamente come i 4 microelaboratori considerati (PPS4 Rockwell, 6800 Motorola, 8080 Intel, 9900 Texas) permettevano di risolvere i vari problemi.
In particolare, Brofferio, ha trattato un'applicazione per un controllo di processo industriale e cioè un regolatore proporzionale di temperatura di un forno, mentre Muzzani ha realizzato un progetto per il controllo di attuatori elettromeccanici in un sistema di stampa (controllo della straslazione della testina di stampa ed avanzamento della carta).
Nel seguito del corso Molinari ha trattato il controllo di unità periferiche, evidenziando come un microelaboratore può essere utilizzato per il governo di una stampante a matrice.
Minnaja ha concluso il ciclo di applicazioni, sviluppando il controllo della trasmissione e della ricezione di un messaggio attraverso una linea di comunicazione seriale, utilizzando un componente specializzato di interfaccia con le unità periferiche.
Ciascuna delle applicazioni presentate è stata sviluppata concretamente nei vari aspetti: dalla struttura fisica del sistema (hardware), all'analisi dei programmi, alla prova di alcune « routines » dei programmi stessi sui sistemi ausiliari di sviluppo dei microelaboratori considerati.

E. R.

La prima rivista italiana di software

È uscito il primo numero di una nuova rivista trimestrale di informatica, pubblicata in collaborazione fra il Centro di Ricerca e Progettazione della Honeywell Information Systems Italia e l'Istituto di Cibernetica dell'Università di Milano.



Si tratta della prima rivista italiana interamente dedicata ai problemi del software. Essa si intitola « Note di software » e intende costituire uno strumento per la rapida e informale circolazione di informazioni, idee ed esperienze nell'area di tali problemi. In particolare essa intende stimolare il dibattito sulle metodologie orientate al miglioramento del rapporto costo/qualità dei programmi, e sugli strumenti atti ad automatizzare l'attività di produzione del software in ogni suo aspetto.
Nel sommario del primo numero figurano, oltre ad alcune rubriche, tre contributi tecnici: « Flagger: uno strumento per la valutazione dei risultati delle prove » di F. Gariboldi, B. Giordani, G. A. Lanzarone, P. V. Renesto; « STNAL: un insieme di macro per la programmazione strutturata in NAL » di C. Poggiali, S. Farnedi, M. Maiocchi, R. Polillo e « Il problema dell'accesso concorrente a una base di dati » di M. Parini.
Fra i programmi della rivista per il prossimo futuro vi è la preparazione di numeri monografici su specifici problemi di software, la segnalazione regolare delle tesi di laurea discusse in Italia su argomenti di software e la pubblicazione di bibliografie - comprendenti oltre a volumi e articoli anche tesi di dottorato nelle università USA - su temi che siano stati segnalati come di particolare interesse dai lettori, coi quali la rivista intende avviare un rapporto decisamente interattivo.
La pubblicazione della nuova rivista (che viene inviata a chi ne faccia richiesta alla redazione, presso la Honeywell Information Systems Italia, 20010 Pregnana Milanese) rientra nell'ambito della collaborazione da tempo in corso fra l'Istituto di Cibernetica dell'Università di Milano e il Centro di Ricerca e Progettazione della Honeywell Information Systems Italia a Pregnana Milanese, dove è in funzione la più grande « software factory » esistente in Italia, e l'unica

specializzata nel software di base degli elaboratori elettronici general purpose.
Tale collaborazione ha dato luogo recentemente alla formulazione di un programma congiunto chiamato « Communication and Programming Project » che ha lo scopo di contribuire alla conoscenza dei problemi connessi con la programmazione degli elaboratori elettronici e, più in generale, con le varie forme di comunicazione uomo-elaboratore ed elaboratore-elaboratore.
Le ricerche che verranno effettuate nell'ambito di questo progetto si articoleranno attorno ai seguenti temi: teoria dei programmi, programmazione strutturata, descrizione di sistemi, strumenti di programmazione, comunicazione uomo-elaboratore (intelligenza distribuita), didattica della programmazione e aspetti di programmazione associati ai sistemi informativi.

G. R.

L'economia del Computer

W. F. Sharpe, Boringhieri, Torino 1976

Il problema di una corretta valutazione economica degli interventi di automazione dei sistemi informativi è lungi dall'essere risolto, anche perché ad esso non sempre s'è prestata la dovuta attenzione.
In effetti, sono gli stessi strumenti concettuali che spesso mancano. È però altrettanto vero che accanto a questa carenza teorica s'è verificata, relativamente alle applicazioni dell'informatica, una « latitanza » manageriale. Anche laddove sono da tempo d'uso corrente tecniche d'analisi economica per la valutazione e il controllo della redditività dell'impiego delle risorse, l'uso dei calcolatori è stato giustificato non raramente sulla base di benefici « intangibili ».
Di certo da qualche tempo, specie in questo periodo di crisi economica, è in atto una revisione critica dei precedenti atteggiamenti verso l'informatica. Ci si sta rendendo conto che, basandosi sulle motivazioni suddette, « intangibili », si è giunti a risultati non sempre soddisfacenti per l'azienda o per l'organizzazione nel suo complesso, e sta diventando sempre più evidente la necessità di un'attenta analisi dell'investimento « calcolatore ».
È proprio la recente pubblicazione della traduzione italiana del volume di W. F. Sharpe, « L'economia del computer », a ben sette anni dall'edizione originale, a denunciare chiaramente il passato disinteresse per gli aspetti economici dell'informatica.
La traduzione dell'opera di Sharpe mette a disposizione di una più ampia cerchia di ricercatori e tecnici, ma soprattutto degli studenti, uno dei pochi classici dell'economia dell'informatica.
Il fondamentale lavoro di Sharpe fu condotto per conto della Rand Corporation tra il 1965 e il '68 nel quadro di un programma di ricerche per l'U.S. Air Force. Si tratta di un autentico trattato di microeconomia applicata all'informatica, fondata su un'analisi sia descrittiva che normativa della domanda e dell'offerta di sistemi di elaborazione.
Purtroppo l'analisi dell'offerta, ad es. delle politiche dei prezzi dei costruttori di calcolatori, delle alternative tra affitto e acquisto, che costituivano un notevole contributo della parte applicativa nell'edizione originale, sono state, più o meno giustamente, soppresse nell'edizione italiana. La ragione è evidente: considerazioni e analisi sulla situazione del mercato americano di dieci anni fa avrebbero avuto ben poco interesse per il lettore italiano d'oggi.
L'edizione italiana così condensata ha trasformato il volume di Sharpe in un manuale, in un corso di microeconomia per informatici.
Degli otto capitoli superstiti (sui dodici originali), cinque costituiscono infatti un sintetico corso di microeconomia, del tutto autoconsistente, che dipanandosi per quasi duecento pagine riprende tutti i principali concetti della teoria microeconomica classica (bastino i titoli dei capitoli: 1) teoria microeconomica, 2) valore e domanda, 3) reddito e profitto, 4) tempo e rischio, 5) costi, fattori produttivi, produzione), illustrati con esempi concernenti la produzione e l'utilizzo dei calcolatori.
Gli ultimi tre capitoli (cento pagine, quanto resta di tutta la parte applicativa presente nell'edizione originale, tre volte più lunga), che costituiscono il cuore dell'opera, sono dedicati all'applicazione dei concetti teorici, esposti nella prima parte, ai concreti problemi economici dell'informatica (6. costo ed efficienza dei sistemi di calcolo; 7. costo ed efficienza della memoria; 8. determinazione dei prezzi dei servizi di calcolo).
Sono proprio alcune pagine di quest'ultima parte che hanno veramente fatto testo: magistrale è, ad esempio, la sintesi sulle economie di scala dei sistemi di elaborazione, dalle origini al 1967. È un peccato che in un libro uscito in Italia nel 1976 non si sia sentito il bisogno di fare il punto, almeno con una nota, sulle economie e diseconomie di scala connesse all'evoluzione tecnologica, hardware e software, successiva.
Altro contributo di Sharpe è quello sul « pricing ». Intanto fa piazza pulita degli equivoci sul fatto se un centro di calcolo debba essere o no un centro di profitto, mostrando chiaramente come la configurazione a centro di profitto del servizio di elaborazione non necessariamente massimizzi il valore netto per l'azienda. Ma soprattutto mostra, coerentemente con la teoria marginalistica, come il problema del pricing debba essere impostato e risolto non in base ai costi ma in base al valore, all'utilità che gli utenti traggono dai servizi di elaborazione. Si sa bene invece come esempi di coerente realizzazione operativa di questa impostazione siano ancora oggi inesistenti, se non in forma bastarda e camuffata (si pensi alla logica ibrida di tutti gli algoritmi di accounting).
Anche queste ultime pagine del libro di Sharpe risentono però del tempo trascorso: tutti i dati, i coefficienti delle equazioni, i parametri empirici derivano e si fondano su una base tecnologica ormai

superata. Ciò non toglie validità sostanziale al discorso e all'impostazione complessiva di Sharpe, ma non si può negare l'esigenza di una riverifica e un aggiornamento dell'imponente lavoro dell'economista americano. Il fatto che non si sia potuto o voluto, almeno per l'edizione italiana, condurre sia pur sommariamente quest'opera di aggiornamento e verifica, dimostra come negli ultimi dieci anni si sia proseguito ben poco, anche a livello internazionale, nel campo dell'« economia del computer ».

Da ultimo, una piccola critica alla traduzione: la persistente traduzione dell'inglese « effectiveness » con il termine italiano « efficienza », anche quando ha chiaramente il senso di « efficacia », rischia di ingenerare non poca confusione nel lettore.

P. M.

La creazione in Francia della fondazione Frederik R. Bull

Nel gennaio scorso è nata a Louveciennes, vicino a Parigi, per iniziativa della CII--Honeywell Bull, la Fondazione Frederik R. Bull. La Fondazione si propone di « contribuire a ogni iniziativa tendente a favorire un rapporto armonico tra l'uomo e lo strumento informatico e a potenziare gli apporti che l'informatica può dare all'ambiente umano ». La Fondazione, che è un'associazione di diritto francese ma aperta a partecipazioni straniere, ammette come membri persone fisiche e persone giuridiche e ha in programma l'organizzazione di un simposio annuale su un tema di carattere generale (quali: informatica e salute; informatica e tempo libero ecc.) nonché di altre manifestazioni periodiche; l'edizione di libri e di una rivista nonché l'organizzazione di un centro di informazioni; la partecipazione a studi e ricerche di informatica sotto forma, tra l'altro, di finanziamenti, di concessione di borse di studio e di messa a disposizione di risorse informatiche; l'attribuzione di premi per incoraggiare e ricompensare lavori e pubblicazioni corrispondenti alle finalità della Fondazione stessa.

Fra le prime realizzazioni della Fondazione figura un libro: « L'enfant et l'ordinateur, ou le secret de Julien », pubblicato in coedizione con Flammarion. Si tratta di un breve racconto, di tono fra didascalico e fiabesco, scritto da Régis Hardy e illustrato da Georges Lemoine, che ha lo scopo di introdurre i bambini nel mondo dell'informatica.

La fondazione, che è presieduta da Maurice Doublet, ex Prefetto di Parigi, si intitola al nome dell'inventore norvegese Frederik Rosing Bull (1882-1925). Impiegato alla Compagnia d'assicurazioni norvegese Störebrand, l'ingegner Bull mise a punto all'inizio degli anni 20 una selezionatrice-addizionatrice a schede perforate, progenitrice di una linea di macchine a schede perforate per la cui fabbricazione e commercializzazione si costituì a Parigi negli anni 30 (dopo la morte dell'inventore, la cui opera venne continuata in particolare dall'altro norvegese Knut A. Knutsen, tuttora vivo, e che è stato chiamato a far parte del Comité de Patronnage della Fondazione) la « Compagnie des Machines Bull ». Da questa, com'è noto, è derivata per successive trasformazioni e fusioni l'attuale CII-Honeywell Bull.

G. R.

Collana dei Quaderni di Informatica

La "Collana dei Quaderni di Informatica" edita dalla Franco Angeli rappresenta la proiezione della rivista sul piano librario. Come la rivista, essa si propone la diffusione in Italia di una "cultura informatica" a livello anche dei non specialisti, ma con una impostazione sempre molto rigorosa e con intenti non puramente didascalico-pratici. Accanto a testi di tipo manualistico, essa pubblica perciò opere di divulgazione ad alto livello, di sintesi su determinati argomenti, di saggistica su temi nuovi e aperti al dibattito.
Della Collana sono usciti finora quattro volumi.

Oggetto di questo libro è l'aspetto hardware: le tecnologie di base, i dispositivi elementari, i blocchi complessi che costituiscono l'elaboratore elettronico.
Il testo è diviso in due parti: nella prima viene presentato, dopo una introduzione su concetti di base, lo stato attuale dell'arte, mentre nella seconda si prospettano gli orientamenti ed i probabili sviluppi avvenire.
Attraverso l'analisi dei fattori critici, vengono fatte previsioni quantitative sulla evoluzione tecnologica dell'elaboratore e se ne valutano i limiti ultimi imposti da leggi fisiche fondamentali.
L'opera è diretta ad una larga fascia di lettori: dagli utilizzatori di elaboratori, agli studenti, a chi, in generale, è interessato a conoscere come queste macchine sono fatte oggi e come potranno esserlo domani.
L'esposizione, pur rigorosa, è stata perciò mantenuta su un livello di larga comprensibilità.

Nato in America alla fine degli anni 50, il Cobol (acronimo di Common Business Oriented Language) è un linguaggio di programmazione evoluto e universale (nel senso che può essere utilizzato praticamente su qualsiasi tipo di elaboratore elettronico) destinato alla risoluzione di problemi gestionali.
La prima parte di quest'opera (la quale fa riferimento alla più recente versione del Cobol, il Cobol ANS 74, che permette di sfruttare le tecniche messe a disposizione dai moderni sistemi operativi) consente di acquisire le nozioni necessarie alla stesura in Cobol di un programma di media difficoltà; nella seconda parte vengono invece trattati esaurientemente problemi relativi alle tabelle ed ai flussi organizzati in modo non sequenziale.
Ogni capitolo è corredato da numerosi esempi e da esercizi in cui vengono presentate alcune delle applicazioni descritte nella parte teorica.

Il Basic, sigla formata dalle iniziali delle parole "Beginners All-purpose Symbolic Instruction Code", è un linguaggio di programmazione "facile": bastano poche ore per apprenderlo e pochi giorni per padroneggiarlo completamente.
Da qui la sua larga diffusione, soprattutto tra i non programmatori.
Questo volume è il primo pubblicato in Italia sul Basic ed è stato particolarmente curato sotto il profilo didattico, al fine di consentire anche l'autoapprendimento del linguaggio.
L'opera è arricchita, poi, da un numero molto elevato di esercizi, per consentire al lettore d'analizzare il significato e il funzionamento di ogni istruzione.

Si tratta di una guida concettuale e operativa alla programmazione strutturata diretta sia a chi già programma con i metodi tradizionali sia a chi per la prima volta si accosta alla programmazione.
Dopo una "introduzione alla programmazione" vengono definiti nei primi capitoli gli strumenti linguistici atti a rappresentare algoritmi, e in particolare viene presentato un insieme flessibile di costrutti di controllo destinati a strutturare i programmi, indicandone per ciascuno il campo di applicabilità e illustrandone l'impiego nell'ambito di alcuni fra i linguaggi oggi più diffusi, quali il Fortran, il Cobol e un assembler tipico.
Nei capitoli successivi viene affrontato il problema della documentazione e della verifica logica del programma, mentre nella parte finale del libro viene esaminata la costruzione di programmi complessi, secondo il metodo della specificazione per successivi livelli di dettaglio.